Circuits, Features, and the Geometry of Meaning

A Mathematical Theory of Mechanistic Interpretability in Neural Networks

Silvano Salvador

Monograph in Mathematical Machine Learning

2026[1cm]

To those who build minds without yet knowing what minds are.

Preface

There is a sense in which modern deep learning has produced two separate intellectual crises at once. The first is practical: we deploy systems whose internal computations are entirely opaque, and this opacity is no longer a curiosity but a liability. The second is scientific: we have before us the most intricate information-processing artifacts ever constructed by human hands, and we have almost no mathematical vocabulary adequate to describe what they are doing. These two crises are not independent. The opacity is itself a symptom of the conceptual poverty.

This monograph is an attempt to begin constructing that vocabulary — not as a catalogue of empirical phenomena, but as a mathematical theory. The subject goes by the name *mechanistic interpretability*: the programme of reverse-engineering neural networks at the level of individual computations, identifying the algorithms they implement and the representations they construct. The field is young, empirically rich, and mathematically underdeveloped. Most of its results consist of case studies — beautiful, surprising, sometimes shocking in their specificity — but assembled without a unifying formal framework. The theorems are few and the definitions are contested.

We propose to do better. The central objects of this theory are *features*, *circuits*, and their geometric interrelations. Features are the representational atoms: directions in activation space that the network uses to encode properties of its inputs. Circuits are the computational molecules: sparse subgraphs of the network that implement identifiable functions by composing features. The geometry of features in high-dimensional space — their mutual angles, the polytopes they form, the frames they span — turns out to govern, with surprising precision, the capacity and cost of the representations the network can construct.

The mathematics we will need is drawn from several traditions: the theory of frames and tight frames from harmonic analysis; compressed sensing and the geometry of sparse recovery; algebraic graph theory and circuit complexity; information-theoretic arguments about capacity and mutual information; and the theory of phase transitions from statistical mechanics. We assume familiarity with linear algebra, basic probability, and the elements of real analysis. We assume no prior knowledge of interpretability.

The ten chapters proceed as follows. Chapters 1–2 establish the static geometry: features as elements of a frame, circuits as subgraphs of a computational DAG. Chapters 3–4 develop the dynamics and information theory: the emergence of circuits via phase transitions during training, and the information bottleneck characterisation of their role. Chapters 5–6 specialise to the transformer architecture and to the phase-transition geometry of feature splitting. Chapters 7–8 address universality and limits: when circuits are the same across independently trained networks, and what cannot, in principle, be mechanistically interpreted. Chapters 9–10 turn to the large-scale regime and to applications: scaling laws as theorems about the asymptotics of circuits, and the implications of the framework for AI safety and alignment.

A word on the ambition and its limits. Mechanistic interpretability is an active and rapidly evolving experimental science. We do not claim to formalise all of it, and we certainly do not claim that every theorem is tight or every definition canonical. What we claim is that the mathematical structures identified here are the *right* ones — that they illuminate rather than

merely accompany the empirical phenomena — and that the theorems we prove, even where incomplete, point toward the questions that matter. The reader who finds gaps is invited to regard them as theorems waiting to be proved rather than as errors.

S.S.
Milan, 2026

How to Use This Monograph

The monograph is designed to be read sequentially, but the following alternative paths may suit different readers.

For the mathematical reader with no background in interpretability. Read Chapters 1–4 (the core mathematical theory), then Chapter 8 (the limits), then the remaining chapters in any order. The key results are Theorems 1.1, 2.8, 3.3, 4.4, and 8.9.

For the machine learning practitioner. Read Chapters 2–3 (circuits and grokking), 5 (attention), and 10 (safety applications), with Chapter 1 (§1.3 on sparsity) as background. The worked examples in §2.8 (IOI circuit) and the phase diagram of §6.5 are the most practically actionable results.

For the AI safety researcher. Read Chapter 4 (information theory), Chapter 7 (universality), Chapter 8 (limits), and Chapter 10 (safety applications) in sequence. Theorem 10.6 (causal faithfulness) and Theorem 8.9 (the IPF) are the central results for safety applications.

For the neuroscientist. Begin with §1.9 (population codes and superposition), Chapter 5 (attention as Hopfield memory), and the bibliographic notes for Chapter 1, which trace the connections to the sparse coding and population coding literature. The convergent evolution discussion in §§7.7 and 10.8 (neuroscience–AI bridge) will be of particular interest.

Prerequisites. We assume familiarity with:

- *Linear algebra*: inner products, norms, eigenvalues, singular values, matrix factorisation;
- *Probability*: random variables, expectations, Gaussian distributions, concentration inequalities;
- *Real analysis*: continuity, differentiability, compactness, the implicit function theorem;
- *Information theory*: entropy, mutual information, KL divergence (at the level of Cover and Thomas, *Elements of Information Theory*).

No prior knowledge of mechanistic interpretability, compressed sensing, or statistical mechanics is assumed.

Acknowledgements

This monograph grew out of lecture notes for a graduate course on mathematical foundations of machine learning at the University of Milan. Many colleagues and students have contributed, through conversations, corrections, and challenges, to sharpening the ideas presented here.

The theory of superposition in Chapter 1 was inspired by the empirical discoveries of the Anthropic mechanistic interpretability team, and particularly by their "Toy Models of Superposition" paper. The connection to frame theory was apparent to anyone working in both fields, but making it mathematically precise required considerable effort; we are grateful to the frame theory community for maintaining the relevant mathematics in an accessible form.

The grokking analysis of Chapter 3 was shaped by conversations with researchers who were, at the time, working on understanding why generalisation sometimes appears so suddenly. The identification of weight decay as the thermodynamic driving force behind grokking emerged from attempting to explain Nanda et al.'s empirical findings in the language of statistical mechanics.

The information-theoretic framework of Chapter 4 builds on a long tradition in machine learning and information theory, but its application to mechanistic interpretability — specifically, the claim that the minimal faithful circuit is the solution to an information bottleneck problem — is, to our knowledge, new. We are grateful to the information theory community for maintaining the relevant results in a form that made this connection possible.

Chapters 8 and 9 were developed in response to the persistent question of what mechanistic interpretability *cannot* do, and what the relationship is between interpretability and the scaling laws. The limits of Chapter 8 were known in principle but had not been stated explicitly in the interpretability context; the scaling law derivations of Chapter 9 were genuinely new.

The safety applications of Chapter 10 reflect the practical urgency that motivates the entire programme. We are grateful to the AI safety community for maintaining a rigorous standard of argument that has pushed us to be more precise about what mechanistic interpretability can and cannot deliver as a safety tool.

Finally, we thank the anonymous reviewers of earlier drafts for their careful reading and incisive questions, many of which are reflected in the open problems of Appendix B.

Contents

Preface **v**

Acknowledgements **vii**

1 The Geometry of Superposition **1**
1.1 The Representational Problem . . . 1
1.2 Frames and the Welch Bound . . . 2
1.3 Sparsity and the Recovery Problem . . . 3
1.4 The Almost-Orthogonality Lemma and Capacity Scaling . . . 3
1.5 Superposition and the Geometry of Polytopes . . . 4
1.6 Beyond Unit Vectors: The Role of Magnitude . . . 4
Bibliographic Notes . . . 5
1.7 Superposition in Practice: Toy Models and Their Lessons . . . 5
1.7.1 The Toy Model Setup . . . 5
1.7.2 The Critical Threshold for Superposition . . . 5
1.7.3 The Geometry of the Optimal Solution . . . 6
1.8 Biological Plausibility and the Population Code . . . 6
1.9 The Relationship Between Superposition and Generalisation . . . 6
Bibliographic Notes . . . 7
1.10 The Connection to Sparse Coding in Neuroscience . . . 7
1.10.1 Sparse Coding as a Special Case . . . 7
1.10.2 The Efficiency of the Neural Code . . . 8
Bibliographic Notes . . . 8
1.11 The Johnson–Lindenstrauss Lemma and Dimensionality Reduction . . . 8
Bibliographic Notes . . . 9

2 Circuits as Mathematical Objects **11**
2.1 The Need for a Formal Notion of Circuit . . . 11
2.2 The Computational Graph and Its Substructure . . . 11
2.3 The Algebra of Circuits . . . 12
2.4 The Circuit Discovery Problem . . . 12
2.5 Faithfulness, Completeness, and the Residual . . . 13
2.6 Transformer Circuits: Attention as a Routing Mechanism . . . 13
2.7 Circuit Complexity and the Depth–Width Tradeoff . . . 14
2.8 The Minimality Problem and Its Hardness . . . 15
Bibliographic Notes . . . 15
2.9 The IOI Circuit: A Complete Worked Example . . . 16
2.9.1 The Task . . . 16
2.9.2 The Circuit Structure . . . 16
2.9.3 Circuit Analysis in Our Formalism . . . 16
2.9.4 The Composition Scores . . . 17

2.10 Circuit Discovery Algorithms in Practice . . . 17
Bibliographic Notes . . . 18
2.11 The Grammar of Computation: Circuit Composition Laws . . . 18
2.11.1 Faithfulness Under Composition . . . 18
2.11.2 Size Under Composition . . . 18
2.11.3 Kolmogorov Complexity Under Composition . . . 19
2.12 The Relationship to Boolean Circuit Complexity . . . 19
2.12.1 Similarities . . . 19
2.12.2 Differences . . . 20
Bibliographic Notes . . . 20
2.13 Circuit Discovery in the Residual Stream Basis . . . 20
2.13.1 The Residual Stream Decomposition . . . 20
2.13.2 Attribution in the Residual Stream . . . 21
2.13.3 Virtual Weights and the Direct Computation . . . 21
Bibliographic Notes . . . 22

3 Grokking and the Phase Transitions of Circuit Emergence 23
3.1 The Phenomenon and Its Paradox . . . 23
3.2 Formal Setup: Grokking as a Transition in Generalisation Gap . . . 23
3.3 Two Regimes and the Hidden Progress Measure . . . 24
3.4 The Norm Gap . . . 24
3.5 The Fourier Analysis of Modular Arithmetic . . . 25
3.6 The Phase Transition: A Free Energy Argument . . . 25
3.7 Generalisation of the Framework . . . 26
3.8 Circuit Crystallisation: A Dynamical Picture . . . 26
Bibliographic Notes . . . 27
3.9 Grokking in Multi-Task Settings . . . 27
3.10 The Grokking Lens on In-Context Learning . . . 28
Bibliographic Notes . . . 28
3.11 The Statistical Mechanics of Gradient Descent . . . 28
3.11.1 The Gradient Flow as a Langevin Process . . . 29
3.11.2 The Free Energy Landscape . . . 29
3.11.3 The Role of Batch Size . . . 30
Bibliographic Notes . . . 30
3.12 Computational Complexity of Grokking Tasks . . . 30
3.12.1 Tasks That Grok Quickly . . . 30
3.12.2 Tasks That Grok Slowly or Not At All . . . 31
3.12.3 The Grokking Frontier . . . 31
Bibliographic Notes . . . 31

4 The Information-Theoretic Duality of Features and Circuits 33
4.1 Two Perspectives on the Same Object . . . 33
4.2 Rate-Distortion Theory: The Fundamental Tradeoff . . . 33
4.3 The Information Bottleneck . . . 34
4.4 Features as Sufficient Statistics . . . 35
4.5 The Circuit–Feature Duality . . . 35
4.6 Mutual Information as a Circuit Attribution Score . . . 36
4.7 The Kolmogorov Complexity Perspective . . . 36
4.8 A Unified Picture . . . 37
Bibliographic Notes . . . 38
4.9 The Variational Information Bottleneck . . . 38

4.10 Information Flow and the Layerwise IB Curve . . . 39
4.11 The Minimal Sufficient Statistic and Its Geometry . . . 39
Bibliographic Notes . . . 40
4.12 Mutual Information Estimation in Practice . . . 40
4.12.1 The Estimation Problem . . . 40
4.12.2 Implications for Circuit Attribution . . . 40
4.12.3 The Faithfulness–Estimation Gap . . . 41
4.13 Gaussian Channels and the Rate-Distortion Bound . . . 41
Bibliographic Notes . . . 41
4.14 Tensor Decomposition and Multi-View Representations . . . 42
4.14.1 Feature Tensors and Their Decomposition . . . 42
4.14.2 Tensor Networks and Deep Circuits . . . 42
4.15 The Dual Representation: Circuits as Linear Operators . . . 43
4.15.1 The Circuit Matrix . . . 43
4.15.2 Non-Linear Circuits and the Tangent Kernel . . . 43
Bibliographic Notes . . . 43
4.16 The Geometry of the Information Bottleneck Tradeoff Curve . . . 44
4.16.1 Convexity and the Shape of the Curve . . . 44
4.16.2 The Circuit Complexity–Performance Curve . . . 44
Bibliographic Notes . . . 44

5 Attention Heads as Associative Memories 45
5.1 The Question Behind the Mechanism . . . 45
5.2 Classical Hopfield Networks and Their Limits . . . 45
5.3 Modern Hopfield Networks: Exponential Capacity . . . 46
5.4 The Exact Correspondence with Attention . . . 46
5.5 Capacity of the Key-Query-Value Architecture . . . 47
5.6 Head Specialisation and the Circuit Perspective . . . 47
5.7 The Sharp vs. Diffuse Attention Regimes . . . 48
Bibliographic Notes . . . 49
5.8 Induction Heads as a Case Study . . . 49
5.9 Multi-Head Attention as a Distributed Associative Memory . . . 50
Bibliographic Notes . . . 50
5.10 Previous Token Heads and the Mechanism of Attention Composition . . . 50
5.10.1 Previous Token Heads . . . 50
5.10.2 The Composition Chain . . . 51
5.10.3 Generalisation: Q-composition and K-composition . . . 51
5.11 From Theory to Practice: Attention Visualisation . . . 51
Bibliographic Notes . . . 52
5.12 Multi-Query Attention and Grouped-Query Attention . . . 52
5.12.1 Multi-Query Attention . . . 52
5.12.2 Grouped-Query Attention . . . 52
Bibliographic Notes . . . 53
5.13 Attention as an Approximate Gradient Descent Step . . . 53
5.13.1 The Energy Gradient . . . 53
5.13.2 Implications for Circuit Depth . . . 54
Bibliographic Notes . . . 54

6 Feature Splitting, Merging, and Polytope Phase Transitions **55**
6.1 The Phenomenon in Full Generality . . . 55
6.2 The Interference Energy Functional . . . 55
6.3 The Stability Analysis: When Configurations Break . . . 56
6.4 Sparsity-Driven Splitting: The Bifurcation Condition . . . 57
6.5 The Full Phase Diagram . . . 57
6.6 Symmetry Breaking and the Order Parameter . . . 58
6.7 Optimal Feature Configurations . . . 58
6.8 Feature Merging: The Reverse Transition and Its Asymmetry . . . 59
Bibliographic Notes . . . 59
6.9 The Landau Theory of Feature Splitting . . . 59
6.9.1 The Landau Expansion . . . 59
6.9.2 The Second-Order Transition . . . 60
6.9.3 Fluctuations and the Ginzburg Criterion . . . 60
6.10 The Geometry of the Feature Moduli Space . . . 60
6.10.1 Connected Components and Their Interpretation . . . 61
6.10.2 The Euler Characteristic of the Moduli Space . . . 61
Bibliographic Notes . . . 61
6.11 Persistent Homology and the Topology of Feature Space . . . 61
6.11.1 The Vietoris-Rips Complex of a Feature Frame . . . 61
6.11.2 Persistent Homology as a Transition Detector . . . 62
6.11.3 Tracking Feature Splitting with Persistent Homology . . . 62
Bibliographic Notes . . . 62
6.12 Numerical Experiments: Toy Models of Splitting . . . 63
6.12.1 Experimental Setup . . . 63
6.12.2 Predicted Phase Boundaries . . . 63
6.12.3 Empirical Validation . . . 63
6.12.4 The Discrete Feature Spectrum . . . 64
6.13 Connections to Random Matrix Theory . . . 64
6.13.1 The Marchenko-Pastur Law . . . 64
6.13.2 Eigenvalue Outliers and Phase Transitions . . . 64
Bibliographic Notes . . . 65
6.14 The Frame Potential and Optimal Configurations . . . 65
6.14.1 The Frame Potential . . . 65
6.14.2 The Learning Dynamics as Frame Potential Minimisation . . . 65
Bibliographic Notes . . . 66

7 Universality Classes of Circuits **67**
7.1 The Universality Question . . . 67
7.2 Defining Circuit Universality . . . 67
7.3 The Platonic Representation Hypothesis . . . 68
7.4 Formal Statement and Proof of Universality . . . 68
7.5 Approximate Universality and the Role of Symmetry . . . 69
7.6 Universality Across Architectures: The Transformer–MLP Equivalence . . . 69
7.7 Convergent Evolution: An Empirical Pattern and Its Explanation . . . 70
7.8 Limits of Universality: When Circuits Diverge . . . 71
Bibliographic Notes . . . 71
7.9 A Worked Example: Universality of Curve Detectors . . . 71
7.10 Universality Across Modalities . . . 72
Bibliographic Notes . . . 72

7.11 Implicit Regularisation and Circuit Selection . . . 73
7.12 Universality and Transfer Learning . . . 73
Bibliographic Notes . . . 74
7.13 Representation Similarity Analysis . . . 74
7.13.1 The Representation Dissimilarity Matrix . . . 74
7.13.2 Centred Kernel Alignment . . . 75
Bibliographic Notes . . . 75
7.14 The Geometry of Convergent Representations . . . 75
7.14.1 The Representation Manifold . . . 75
7.14.2 The Convergence Basin . . . 76
7.14.3 Measuring Convergence in Practice . . . 76
Bibliographic Notes . . . 76

8 The Limits of Mechanistic Interpretability 77
8.1 What Cannot Be Interpreted? . . . 77
8.2 The Neural Network Verification Problem and Its Undecidability . . . 77
8.3 The Complexity of Approximate Interpretability . . . 78
8.4 Statistical Limits: The Sample Complexity of Mechanistic Claims . . . 79
8.5 Irreducible Complexity and Non-Decomposable Circuits . . . 79
8.6 The Interpretability–Performance Tradeoff . . . 80
8.7 What the Limits Do and Do Not Preclude . . . 80
8.8 The Horizon Problem: Interpreting Interpreters . . . 81
Bibliographic Notes . . . 81
8.9 Logical Depth and the Complexity of Understanding . . . 82
8.10 Approximate Circuits and Robust Interpretability . . . 82
Bibliographic Notes . . . 83
8.11 The Verification Gap and Its Implications . . . 83
8.11.1 Specification vs. Verification . . . 83
8.11.2 The Role of Human Verification . . . 83
8.11.3 Compositional Verification as a Partial Solution . . . 84
8.12 Adversarial Robustness Through the Interpretability Lens . . . 84
Bibliographic Notes . . . 85
8.13 The Complexity of Circuit Equivalence . . . 85
8.13.1 Functional Equivalence . . . 85
8.13.2 The Isomorphism Gap . . . 86
8.13.3 Approximate Isomorphism and Its Complexity . . . 86
8.14 Decidability for Special Classes . . . 86
8.14.1 Linear Circuits . . . 86
8.14.2 Single-Layer ReLU Circuits . . . 87
Bibliographic Notes . . . 87
8.15 Quantitative Bounds on the Interpretability Gap . . . 87
8.15.1 The Effective Circuit Size and Sample Complexity . . . 87
8.15.2 The Practical Approximation Ratio . . . 87
8.15.3 The Verification Gap in Numbers . . . 88
Bibliographic Notes . . . 88

9 Scaling Laws and the Asymptotic Theory of Circuits 89
9.1 Scaling as a Mathematical Phenomenon . . . 89
9.2 Feature Importance Distributions and Their Tails . . . 89
9.3 Deriving the Parameter Scaling Exponent . . . 90
9.4 Circuits at Scale: The Asymptotic Regime . . . 91

9.5 Emergent Capabilities and Phase Transitions at Scale . . . 91
9.6 The Double Descent Phenomenon in the Circuit Framework . . . 91
Bibliographic Notes . . . 92
9.7 Optimal Compute Allocation: The Chinchilla Frontier . . . 92
9.8 Connections to Statistical Physics . . . 93
Bibliographic Notes . . . 93
9.9 Scaling Laws for Interpretability . . . 93
9.10 Connections to Algorithmic Information Theory . . . 94
Bibliographic Notes . . . 95
9.11 The Sample Complexity of Feature Learning . . . 95
9.11.1 The Required Sample Complexity . . . 95
9.11.2 The Cumulative Learning Curve . . . 95
9.11.3 The Interplay Between Model Capacity and Data . . . 96
9.12 Feature Decomposition and the Semantic Hierarchy . . . 96
Bibliographic Notes . . . 96
9.13 Mechanistic Interpretability and the Scaling Hypothesis . . . 97
9.13.1 The Strong Scaling Hypothesis . . . 97
9.13.2 The Interpretability Scaling Hypothesis . . . 97
9.13.3 Implications for the Safety Research Agenda . . . 97
Bibliographic Notes . . . 98
9.14 The Phase Diagram of Scaling . . . 98
9.14.1 Three Scaling Regimes . . . 98
Bibliographic Notes . . . 98

10 Applications to AI Safety and Alignment **99**
10.1 The Safety Imperative . . . 99
10.2 Detecting Deception: The Circuit Signature of Dishonesty . . . 99
10.3 Alignment Failures as Circuit Misalignment . . . 100
10.4 The Representation Geometry of Truthfulness . . . 101
10.5 Formalising Alignment as a Circuit Design Problem . . . 101
10.6 Scalable Oversight and the Circuit Hierarchy . . . 102
10.7 The Geometry of Values . . . 102
10.8 Coda: The Shape of the Theory . . . 103
Bibliographic Notes . . . 104
10.9 Mechanistic Interpretability as a Scientific Programme . . . 104
10.9.1 The Levels of Analysis . . . 104
10.9.2 Explanatory vs. Predictive Interpretability . . . 104
10.9.3 The Hypothesis-Driven Approach . . . 105
10.10 A Research Agenda . . . 105
Bibliographic Notes . . . 106
10.11 Mechanistic Red-Teaming . . . 106
10.11.1 Circuit-Based Mitigation Strategies . . . 107
10.12 The Neuroscience–AI Interpretability Bridge . . . 107
10.12.1 Shared Mathematical Tools . . . 107
10.12.2 Shared Empirical Findings . . . 108
Bibliographic Notes . . . 108
10.13 Interpretability-Based Anomaly Detection . . . 108
10.13.1 Circuit Activation Anomalies . . . 108
10.13.2 Feature-Level Anomaly Detection . . . 109
Bibliographic Notes . . . 109

11 Synthesis: The Mathematical Architecture of Understanding **111**
11.1 The Three Fundamental Dualities . . . 111
11.1.1 The Feature–Circuit Duality . . . 111
11.1.2 The Geometry–Information Duality . . . 111
11.1.3 The Dynamics–Complexity Duality . . . 112
11.2 The Interpretability Landscape . . . 112
11.2.1 The Axes of the Frontier . . . 112
11.2.2 Where Models Sit on the Frontier . . . 113
11.3 The Role of Sparsity in the Theory . . . 113
11.4 Connections to Adjacent Fields . . . 114
11.4.1 Theoretical Machine Learning . . . 114
11.4.2 Statistical Mechanics . . . 114
11.4.3 Harmonic Analysis . . . 114
11.4.4 Algebraic Geometry and Representation Theory . . . 114
11.5 What the Theory Achieves . . . 115
11.5.1 What Has Been Achieved . . . 115
11.5.2 What Has Not Been Achieved . . . 115
11.6 A Final Reflection . . . 116
Bibliographic Notes . . . 116
11.7 The Future of the Theory: Open Directions . . . 116
11.7.1 A Theory of Feature Semantics . . . 116
11.7.2 The Dynamics of Feature Learning . . . 117
11.7.3 Non-Linear Extensions . . . 117
11.7.4 The Theory in the Age of Foundation Models . . . 117
11.8 Philosophical Coda: What It Means to Understand a Mind . . . 118
Bibliographic Notes . . . 119
11.9 Bridges to Classical Mathematics . . . 119
11.9.1 Convex Geometry and the Geometry of Features . . . 119
11.9.2 Algebraic Topology and Circuit Homotopy . . . 119
11.9.3 Harmonic Analysis on Groups and Grokking . . . 120
11.9.4 Statistical Decision Theory and Optimality . . . 120
11.10 Selected Research Problems in Detail . . . 121
11.10.1 Problem 1: Deriving the Chinchilla Exponents . . . 121
11.10.2 Problem 3: Universality Beyond IB Optimality . . . 121
11.10.3 Problem 5: Causal Faithfulness at Scale . . . 121
Bibliographic Notes . . . 122
11.11 The Information-Geometric Perspective on the Theory . . . 122
11.11.1 The Fisher-Rao Metric on Representations . . . 122
11.11.2 The Natural Gradient and Circuit Learning . . . 123
11.11.3 Connections to the IB Objective . . . 123
11.12 A Research Programme for the Theory's Continuation . . . 123
11.12.1 The Five Core Theorems to Be Proved . . . 124
11.12.2 The Three Core Experiments to Be Run . . . 124
Bibliographic Notes . . . 124

A Collected Proofs and Technical Complements **127**
A.1 Proofs for Chapter 1 . . . 127
A.1.1 The Welch Bound in Full Generality . . . 127
A.1.2 Proof of the RIP for Random Frames (Theorem 1.5) . . . 127
A.1.3 Concentration of Measure on the Sphere . . . 128

A.1.4 Optimal Magnitude Allocation (Proposition 1.8) . . . 128
A.2 Proofs for Chapter 2 . . . 128
A.2.1 Hardness of Minimum Circuit (Theorem 2.8) . . . 128
A.3 Proofs for Chapter 3 . . . 129
A.3.1 The Norm Gap (Lemma 3.2) . . . 129
A.3.2 The Fourier Structure of Grokked Networks . . . 129
A.4 Proofs for Chapter 4 . . . 129
A.4.1 MDL Bound on Faithful Circuit Size (Proposition 4.7) . . . 129
A.4.2 Proof of the Circuit–Feature Duality (Theorem 4.4) . . . 130
A.5 Proofs for Chapter 5 . . . 130
A.5.1 Modern Hopfield Capacity (Theorem 5.2) . . . 130
A.5.2 Attention Entropy Critical Temperature (Proposition 5.6) . . . 130
A.6 Proofs for Chapter 6 . . . 131
A.6.1 The Gradient Flow on the Frame Manifold (Appendix A) . . . 131
A.6.2 Symmetry Breaking and Mean-Field Exponents (§6.6) . . . 131
A.7 Proofs for Chapter 8 . . . 131
A.7.1 Sample Complexity of Mechanistic Verification (Theorem 8.5) . . . 131
A.7.2 Shape of the IPF (Theorem 8.9) . . . 132

B Open Problems **133**
B.1 Proofs for Chapter 7 . . . 134
B.1.1 Transformer–MLP Equivalence (Theorem 7.6) . . . 134
B.1.2 Gradient Flow Circuit Selection (Proposition 7.10) . . . 135
B.2 Proofs for Chapter 9 . . . 135
B.2.1 Optimal Parameter–Data Tradeoff (Proposition 9.8) . . . 135
B.2.2 Interpretability Scaling Law (Proposition 9.9) . . . 135
B.3 Additional Results: The Information Geometry of Circuits . . . 135
B.4 Proofs for Chapter 10 . . . 136
B.4.1 Circuit-Based Deception Detection (Theorem 10.2) . . . 136
B.4.2 Causal Faithfulness (Theorem 10.6) . . . 136
B.5 Proofs for Chapter 11 . . . 137
B.5.1 Coding Efficiency and the Welch Bound (Proposition 1.10) . . . 137
B.5.2 Superposition and Generalisation (Proposition 1.9) . . . 137
B.5.3 Feature Accuracy and Sample Complexity (Equation 9.13) . . . 137
B.6 A Worked Example: The Modular Addition Circuit in Full . . . 137
B.6.1 Setup . . . 137
B.6.2 The Features . . . 137
B.6.3 The Circuit . . . 138
B.6.4 Faithfulness . . . 138
B.6.5 Information-Theoretic Analysis . . . 138
B.6.6 Kolmogorov Complexity . . . 138
B.7 Additional Proofs for Chapter 11 . . . 139
B.7.1 Fisher Information and the Welch Bound (Proposition 11.3) . . . 139
B.7.2 Anomaly Detection and Distribution Shift (Proposition 10.12) . . . 139
B.7.3 Grokking Condition (Proposition 3.7) . . . 139
B.8 A Complete Index of Main Results . . . 139
B.8.1 Theorems . . . 140
B.8.2 Definitions . . . 141
B.8.3 Hypotheses . . . 141

C Notation and Mathematical Conventions **143**
C.1 General Conventions . . . 143
C.2 Linear Algebra . . . 143
C.3 Probability and Statistics . . . 144
C.4 Information Theory . . . 144
C.5 Frame Theory . . . 144
C.6 Neural Network Architecture . . . 145
C.7 Transformer Attention . . . 145
C.8 Interpretability Framework . . . 145
C.9 Asymptotics and Order Notation . . . 146
C.10 Scaling Law Parameters . . . 146
C.11 On Mathematical Rigour . . . 146
C.12 Glossary of Technical Terms . . . 147

D Annotated Bibliography **153**
D.1 Frame Theory and Compressed Sensing (Chapter 1) . . . 153
D.2 Circuit Theory (Chapter 2) . . . 154
D.3 Grokking and Phase Transitions (Chapter 3) . . . 154
D.4 Information Theory of Circuits (Chapter 4) . . . 155
D.5 Attention Heads as Associative Memories (Chapter 5) . . . 155
D.6 Feature Splitting and Phase Transitions (Chapter 6) . . . 155
D.7 Universality (Chapter 7) . . . 156
D.8 Limits of Interpretability (Chapter 8) . . . 156
D.9 Scaling Laws (Chapter 9) . . . 157
D.10 AI Safety and Alignment (Chapter 10) . . . 157

Chapter 1

The Geometry of Superposition

The aim of science is not to open a door to infinite wisdom, but to set a limit to infinite error.

Bertolt Brecht, *Life of Galileo*

1.1 The Representational Problem

Any attempt to understand what a neural network is doing must begin with a question that seems deceptively simple: what does a neuron represent? In the classical picture, inherited from computational neuroscience, a single neuron corresponds to a single concept — the grandmother cell of cognitive science folklore. This picture is flatly inconsistent with the empirical behaviour of large language models and vision transformers, where individual neurons consistently respond to multiple semantically unrelated stimuli, and where individual concepts are consistently encoded across multiple neurons. The phenomenon goes by the name *polysemanticity*, and it is not a bug. It is, as we will argue, a mathematically optimal response to a representational constraint.

To state the constraint precisely, let us establish notation. Let $\mathcal{H} = \mathbb{R}^d$ be the *representation space* at some intermediate layer of a neural network — d is the number of neurons. The network is exposed to a world that contains m binary or continuously-valued *features*: properties of inputs that matter for the tasks the network must perform. We write $f_1, \ldots, f_m \in S^{d-1}$ for the unit vectors in $\mathcal{H}$ that the network uses to represent these features, where $S^{d-1} = \{v \in \mathbb{R}^d : \|v\| = 1\}$ is the unit sphere. The *activation* of feature i on input x is a scalar $a_i(x) \geq 0$, and the *representation* of x in $\mathcal{H}$ is the superposition

$$h(x) = \sum_{i=1}^{m} a_i(x)\, f_i. \tag{1.1}$$

This is not an assumption about how the network organises its internal representations; it is a constraint that any linear readout of the hidden state must respect. If a downstream neuron v reads out feature i via the weight vector f_i, its activation is

$$f_i^\top h(x) = a_i(x) + \sum_{j \neq i} a_j(x)(f_i^\top f_j).$$

The second term is *interference* — the ghost of other features bleeding into the read-out of feature i. When $m \leq d$ and the feature vectors are orthogonal, interference vanishes, and every feature can be read out perfectly. But this is impossible when $m > d$, which is the regime of genuine interest.

The representational problem is therefore: how should one choose m feature vectors in $\mathbb{R}^d$, with $m \gg d$, so that the interference is small and the features can be recovered approximately? This question has a long and distinguished history in approximation theory, frame theory, and compressed sensing. We will discover that the network's empirical answer — polysemantic neurons, superposed representations — is essentially the same as the mathematical one.

1.2 Frames and the Welch Bound

The right language for discussing collections of vectors that span a space while possibly being far more numerous than the dimension is the language of *frames*. Recall that a frame for $\mathbb{R}^d$ is a sequence $\{f_i\}_{i=1}^m$ of vectors satisfying

$$A\,\|v\|^2 \;\le\; \sum_{i=1}^{m} |\langle v,\, f_i\rangle|^2 \;\le\; B\,\|v\|^2 \qquad \forall\, v \in \mathbb{R}^d \tag{1.2}$$

for constants $0 < A \le B < \infty$ called the *frame bounds*. When $A = B$, the frame is *tight*; when additionally $\|f_i\| = 1$ for all i, it is a *unit-norm tight frame* (UNTF). Tight frames are the overcompleteness analogues of orthonormal bases: they provide a perfectly stable decomposition of any vector in $\mathbb{R}^d$ into m coefficients, at the cost of redundancy.

The frame operator $S : \mathbb{R}^d \to \mathbb{R}^d$ is defined by $Sv = \sum_{i=1}^m \langle v,\, f_i\rangle f_i$. For a tight frame with bound A, one has $S = AI_d$, which immediately gives the reconstruction formula $v = \frac{1}{A}\sum_i \langle v,\, f_i\rangle f_i$. The synthesis matrix $F = [f_1 \mid \cdots \mid f_m] \in \mathbb{R}^{d\times m}$ satisfies $S = FF^\top$, and for a unit-norm tight frame one has $A = m/d$.

The key quantity governing the quality of superposed representations is the *coherence*:

$$\mu(F) \;=\; \max_{i\neq j} |\langle f_i,\, f_j\rangle|\,.$$

Small coherence means the feature vectors are nearly orthogonal; large coherence means heavy interference.

Theorem 1.1 (Welch Bound [66])**.** *For any collection of m unit vectors $f_1, \dots, f_m$ in $\mathbb{R}^d$ with $m > d$,*

$$\mu(F) \;\ge\; \sqrt{\frac{m-d}{d(m-1)}}. \tag{1.3}$$

Proof. Let $G = F^\top F \in \mathbb{R}^{m\times m}$. The diagonal entries are $G_{ii} = 1$, so $\operatorname{tr}(G) = m$. Since $\operatorname{rank}(G) \le d$, by the Cauchy–Schwarz inequality applied to the at most d nonzero eigenvalues:

$$\sum_{i=1}^{d} \lambda_i^2 \;\ge\; \frac{(\operatorname{tr} G)^2}{d} = \frac{m^2}{d}.$$

Since $\|G\|_F^2 = m + \sum_{i\neq j} G_{ij}^2 \ge m^2/d$, we get $\sum_{i\neq j} G_{ij}^2 \ge m(m-d)/d$. Dividing by the $m(m-1)$ off-diagonal pairs gives the bound. □

The Welch bound is achieved by *equiangular tight frames* (ETFs), which exist for many pairs (d, m) and connect to finite geometry, Hadamard matrices, and strongly regular graphs. The *absolute bound* gives a necessary condition: $m \le d(d+1)/2$ for real frames.

For the representational problem of neural networks, Theorem 1.1 has a concrete implication: with m features in $\mathbb{R}^d$, the maximum pairwise interference cannot be reduced below approximately $\sqrt{m/d^2}$ when $m \gg d$. The price of superposition scales as $\sqrt{m/d^2}$.

1.3 Sparsity and the Recovery Problem

The Welch bound establishes a fundamental lower bound on interference but says nothing about the conditions under which features can be recovered in spite of it. The recovery criterion is: given $h = Fa$, reconstruct the coefficient vector a. Since $m > d$, this is underdetermined. Recovery is possible only when we exploit *sparsity*.

Definition 1.2. A vector $a \in \mathbb{R}^m$ is *k-sparse* if $\|a\|_0 \le k$. Its support is $\operatorname{supp}(a) = \{i : a_i \neq 0\}$.

In the neural network context, sparsity means that on any given input, only a small fraction of features are active. With this structure, the recovery problem becomes: find the sparsest a satisfying $Fa = h$. The ℓ^0 formulation is NP-hard, but the central miracle of compressed sensing is that the ℓ^1 relaxation

$$\hat{a} \;=\; \arg\min_{a'} \|a'\|_1 \quad \text{subject to} \quad Fa' = h \tag{1.4}$$

recovers the true sparse a exactly, under the Restricted Isometry Property.

Definition 1.3 (Restricted Isometry Property)**.** The matrix $F \in \mathbb{R}^{d \times m}$ satisfies RIP of order k with constant $\delta_k \in (0, 1)$ if

$$(1 - \delta_k)\,\|a\|^2 \;\le\; \|Fa\|^2 \;\le\; (1 + \delta_k)\,\|a\|^2 \tag{1.5}$$

for every k-sparse $a \in \mathbb{R}^m$.

Theorem 1.4 (Exact Recovery [15, 21])**.** *If F satisfies RIP of order $2k$ with $\delta_{2k} < \sqrt{2} - 1 \approx 0.414$, then for every k-sparse $a^* \in \mathbb{R}^m$, the solution to (1.4) with $h = Fa^*$ satisfies $\hat{a} = a^*$.*

RIP of order k is achievable with $d = O(k \log(m/k))$ measurements. This is the theoretical basis for superposition: d neurons can faithfully represent up to k simultaneously active features from a dictionary of $m \gg d$ total features, provided $d \gtrsim k \log(m/k)$.

Theorem 1.5 (Random Frames Satisfy RIP)**.** *Let the columns of $F \in \mathbb{R}^{d \times m}$ be drawn independently and uniformly from S^{d-1}. Then F satisfies RIP of order k with constant $\delta_k \le \varepsilon$ with probability at least $1 - 2e^{-c\varepsilon^2 d}$, provided $m \le e^{c'\varepsilon^2 d/k}$ for absolute constants $c, c' > 0$.*

A randomly chosen dictionary of $m = e^{O(d/k)}$ features in $\mathbb{R}^d$ supports perfect recovery of any k-sparse activation pattern. The number of representable features grows exponentially in d/k. If $k = 10$ and $d = 1000$, a network can represent e^{100} distinct features in a 1000-dimensional space.

1.4 The Almost-Orthogonality Lemma and Capacity Scaling

Lemma 1.6 (Almost-Orthogonality Capacity)**.** *For any $\varepsilon > 0$, there exist $m = e^{\Omega(\varepsilon^2 d)}$ unit vectors $f_1, \ldots, f_m \in S^{d-1}$ with $|\langle f_i,\, f_j \rangle| \le \varepsilon$ for all $i \neq j$.*

Proof. Draw m vectors independently and uniformly from S^{d-1}. For any fixed pair (i, j), the concentration of measure inequality on S^{d-1} [40] gives:

$$\mathbb{P}[|\langle f_i,\, f_j \rangle| \ge \varepsilon] \;\le\; 2e^{-\varepsilon^2 d/2}.$$

By the union bound, all $\binom{m}{2} \le m^2/2$ pairs satisfy $|\langle f_i,\, f_j \rangle| \le \varepsilon$ simultaneously with probability at least $1 - m^2 e^{-\varepsilon^2 d/2}$, which is positive provided $m \le e^{\varepsilon^2 d/4}$. □

This exponential capacity is the mathematical heart of superposition. In a layer of dimension $d = 4096$ with tolerated interference $\varepsilon = 0.1$, Lemma 1.6 guarantees approximately 10^{17} nearly-orthogonal feature directions.

Proposition 1.7 (Capacity–Interference Tradeoff)**.** *Consider a network representing m features in $\mathbb{R}^d$ with maximum pairwise interference $\varepsilon = \mu(F)$ and maximum instantaneous sparsity k. The expected squared read-out error for feature i under a uniform distribution over k-sparse activation patterns satisfies:*

$$\mathbb{E}[\text{error}_i^2] \;\leq\; k\varepsilon^2. \tag{1.6}$$

The maximum number of features achievable with interference ε is $m \leq e^{O(\varepsilon^2 d)}$, so $\log m \lesssim \varepsilon^2 d$.

A network that wants to store many features must accept proportionally more interference, tolerable only when activations are sparse.

1.5 Superposition and the Geometry of Polytopes

Consider $m = d+1$ features in $\mathbb{R}^d$. By Theorem 1.1, the minimum coherence is $1/d$, achieved when the $d+1$ unit vectors form the vertices of a regular simplex centred at the origin. For $d = 2$ and $m = 3$, three unit vectors at 120° to each other. For $d = 3$ and $m = 4$, the vertices of a regular tetrahedron inscribed in S^2.

More generally, the optimal packing of m unit vectors in $\mathbb{R}^d$ — minimising maximum coherence — corresponds to centred regular polytopes when $m \leq d+1$, and to the vertices of more complex geometric objects when $m > d+1$. What is striking, and confirmed empirically by [24], is that the feature configurations learned by neural networks in the superposition regime closely resemble these optimal geometric configurations. When $m = 2d$, the optimal configurations are related to the vertices of cross-polytopes. Networks trained with sparsity-inducing regularisation approximate these geometric optima.

This perspective clarifies *feature splitting*: as the sparsity prior strengthens, networks decompose a polysemantic neuron into two monosemantic neurons. In geometric language, the feature vector splits from a single shared vertex (interference ≈ 1) into two distinct orthogonal vertices (interference ≈ 0). We return to the mathematics of such phase transitions in Chapter 6.

1.6 Beyond Unit Vectors: The Role of Magnitude

Trained networks do not obey the unit-norm constraint: the feature vectors f_i have varying norms encoding importance. Let $f_i = r_i \hat{f}_i$ where $r_i = \|f_i\|$ is the magnitude. The signal-to-noise ratio for reading out feature i is approximately:

$$\text{SNR}_i \;\approx\; \frac{r_i a_i}{k\,\bar{r}\,\bar{a}\,\varepsilon},$$

where $\bar{r}$ and $\bar{a}$ are averages of other features' magnitudes and activations. Features with larger r_i have better SNR.

Proposition 1.8 (Optimal Magnitude Allocation)**.** *Consider m features in $\mathbb{R}^d$ with fixed directions forming an ETF, and fixed total energy $\sum_i r_i^2 = E$. Suppose feature i is active with probability p_i and contributes squared loss r_i^{-2} when misread. The expected total loss is minimised by setting $r_i \propto p_i^{1/4}$.*

This is consistent with mechanistic interpretability observations: features corresponding to high-frequency or high-importance concepts have larger weight norms.

Bibliographic Notes

The superposition hypothesis is developed in [24]. The mathematical theory of frames is treated in [17]. The Welch bound is from [66]. The foundational theorems of compressed sensing are in [15] and [21]. Concentration of measure on the sphere is in [40]. ETFs and their connection to finite geometry are surveyed in [16] and [25].

1.7 Superposition in Practice: Toy Models and Their Lessons

The theoretical framework developed above becomes concrete through the toy models introduced by Elhage et al. [24]. We briefly describe these models and interpret their behaviour through the lens of the Welch bound and the capacity–interference tradeoff.

1.7.1 The Toy Model Setup

Consider a network with a single bottleneck hidden layer of dimension $d = 1$ (a single neuron) and m input features, each of which is binary: $a_i(x) \in \{0, 1\}$ with $\mathbb{P}[a_i = 1] = p_i$ and the activations assumed independent. The importance of feature i is modelled as $\lambda_i > 0$, with features ordered so that $\lambda_1 \geq \cdots \geq \lambda_m$. The network encodes the m-dimensional input into a single real number $h = \sum_i a_i r_i$ (using the scalar version of (1.1) with $d = 1$), and the loss is the mean squared error between the reconstructed activations and the true activations:

$$\mathcal{L} \;=\; \sum_{i} \lambda_i \, \mathbb{E}\big[(a_i - \hat{a}_i)^2\big] \,, \tag{1.7}$$

where $\hat{a}_i$ is the reconstruction of feature i from h.

1.7.2 The Critical Threshold for Superposition

With $d = 1$, the Welch bound (1.3) gives $\mu(F) \geq \sqrt{(m-1)/1 \cdot (m-1)} = 1$ for $m \geq 2$: all pairwise inner products are ± 1 when $d = 1$. This means any two features in the same dimension must be encoded on the same direction (with the same or opposite sign), leading to maximal interference. Despite this, the optimal strategy for a sufficiently sparse distribution is superposition: encode each feature as $f_i = r_i \in \{-1, +1\}$ (all features use the same direction with possibly flipped sign), and rely on sparsity to make simultaneous activations rare.

For the toy model with $m = 2$ and $\lambda_1 = \lambda_2 = \lambda$, $p_1 = p_2 = p$, the expected loss under superposition (both features encoded) is:

$$\mathcal{L}_{\text{super}} \;=\; 2\lambda p(1-p) \cdot 1^2 \;=\; 2\lambda p(1-p), \tag{1.8}$$

because the only source of error is simultaneous activation of both features (probability p^2 for the product), each causing an error of magnitude 1. The expected loss under monosemantic encoding (only feature 1 encoded) is:

$$\mathcal{L}_{\text{mono}} \;=\; \lambda \cdot p \cdot 1 \;=\; \lambda p, \tag{1.9}$$

because feature 2 is never reconstructed (error always 1) with probability p. Superposition is preferred when $\mathcal{L}_{\text{super}} < \mathcal{L}_{\text{mono}}$, i.e., when $2p(1-p) < p$, i.e., when $p < 1/2$. For features with activation probability less than $1/2$ — which is the case for most features in natural data — superposition is the optimal strategy even in $d = 1$.

1.7.3 The Geometry of the Optimal Solution

For $m > 2$ features and $d < m$, the optimal feature vectors take a characteristic geometric form. When $m = d + 1$, the optimum is the regular simplex (Proposition 6.6(2)): all features are encoded as unit vectors equally spaced around the sphere. When $m = 2d$, the optimum is the cross-polytope: d pairs of antipodal features. These are precisely the configurations that achieve the Welch bound (Theorem 1.1).

The toy models of [24] observe exactly this pattern: when m/d is a small integer (such as 1.5 or 2), the network learns configurations closely approximating the regular polytopes predicted by Proposition 6.6. When m/d is larger (say 5 or 10), the network learns approximately random frames, consistent with the exponential capacity of Lemma 1.6. The transition between these regimes occurs near the critical sparsity k^* of Theorem 6.4.

1.8 Biological Plausibility and the Population Code

The theory of superposition described in this chapter has an interesting antecedent in computational neuroscience: the theory of *population codes*. In neuroscience, it has long been known that individual neurons are tuned to respond to multiple stimuli (they are "polysemantic" in the terminology of this monograph), and that individual stimuli activate multiple neurons. The representation of a stimulus is distributed across a population of neurons, and the stimulus is decoded by reading out the population activity as a whole.

The mathematical framework of population codes, developed by Seung and Sompolinsky [56], is essentially the same as the superposition framework of this chapter: each neuron j has a "tuning curve" $f_j(s)$ describing its response to stimulus s, and the representation of s in the population is the vector $(f_1(s), \ldots, f_d(s)) \in \mathbb{R}^d$. The read-out of any property $g(s)$ is a linear combination $\sum_j w_j f_j(s)$, and the accuracy of the read-out is determined by the Fisher information of the population code — a quantity that, for Poisson-distributed spike counts, is proportional to the squared coherence $\mu(F)^2$ of the tuning curves.

The Welch bound (Theorem 1.1) therefore has a direct neural counterpart: the minimum Fisher information of any population code for m stimuli in d neurons is bounded below by a quantity proportional to $(m - d)/[d(m - 1)]$. This is the fundamental limit of population coding, analogous to the capacity–interference tradeoff of Proposition 1.7.

1.9 The Relationship Between Superposition and Generalisation

We close this chapter by connecting superposition to the learning-theoretic notion of generalisation. The capacity–interference tradeoff (Proposition 1.7) bounds the expected read-out error on training inputs. But does a network that achieves low training error via superposition also generalise to test inputs?

The answer depends on the geometry of the feature space. A network that has learned a frame F that achieves the Welch bound (Theorem 1.1) will, by the RIP of Theorem 1.5, correctly decode any k-sparse combination of features — not just the combinations seen during training, but any combination that arises on test inputs. This is the sense in which achieving the Welch bound is a *sufficient condition* for generalisation: it guarantees that the network's representation supports arbitrary sparse read-outs, regardless of which features are active on any given test input.

Proposition 1.9 (Superposition and Generalisation)**.** *Let F be a feature frame satisfying the RIP of order $2k$ with constant $\delta_{2k} < \sqrt{2} - 1$. Then for any k-sparse test input a^*:*

$$\|\hat{a} - a^*\| \;\leq\; C\delta_{2k}\,\|a^*\|\,, \tag{1.10}$$

where $\hat{a}$ is the ℓ^1-recovered activation vector and $C > 0$ is a universal constant.

This proposition follows directly from Theorem 1.4 applied to the test input. It says that a network with a RIP-satisfying frame generalises to test inputs: the reconstruction error is bounded by the RIP constant times the signal norm, regardless of the specific test input.

Bibliographic Notes

The superposition hypothesis and toy models are developed in [24]. Frame theory is treated in [17]. The Welch bound is from [66]. Compressed sensing is from [15] and [21]. Concentration of measure is in [40]. ETFs are surveyed in [16] and [25]. Population codes and the Fisher information connection are from Seung and Sompolinsky [56].

1.10 The Connection to Sparse Coding in Neuroscience

The theory of superposition developed in this chapter has deep roots in the neuroscience literature on *sparse coding* [48]. The sparse coding framework, introduced to explain the receptive fields of simple cells in the primary visual cortex, proposes that neurons represent sensory inputs as sparse linear combinations of basis functions. This is mathematically identical to the superposition hypothesis.

1.10.1 Sparse Coding as a Special Case

In the sparse coding framework of Olshausen and Field [48], the visual input $x \in \mathbb{R}^n$ (a patch of an image) is represented as a sparse linear combination of *basis functions* $\phi_1, \ldots, \phi_m \in \mathbb{R}^n$:

$$x \;=\; \sum_{i=1}^{m} a_i \phi_i + \varepsilon, \tag{1.11}$$

where $a = (a_1, \ldots, a_m)^\top$ is a sparse activation vector and ε is noise. The basis functions are learned by minimising the reconstruction error subject to a sparsity constraint on the activations:

$$\min_{\{\phi_i\},a} \left\| x - \sum_i a_i \phi_i \right\|^2 + \lambda \|a\|_1 \,. \tag{1.12}$$

This is exactly the superposition framework of (1.1) applied in the *input space* rather than the *representation space*: the features ϕ_i span the input x rather than being directions in a hidden layer h. The ℓ^1 sparsity penalty in (1.12) is the same as the ℓ^1 recovery criterion of Theorem 1.4.

The key result of sparse coding is that the learned basis functions ϕ_i, when trained on natural images, spontaneously develop into oriented Gabor wavelets — localised, oriented, and bandpass — closely matching the receptive fields of simple cells in the primary visual cortex. This is the neural network community's oldest example of emergent feature learning: the network discovers the optimal basis for representing natural images, and this basis happens to match the biological visual system's solution.

1.10.2 The Efficiency of the Neural Code

The sparse coding framework connects to the Welch bound and the capacity-interference tradeoff through the concept of *coding efficiency*. Define the efficiency of a sparse code as the ratio of the mutual information $I(X;T)$ (information about the input captured by the representation) to the total neural activity $\sum_i \mathbb{E}[a_i]$ (the metabolic cost of the representation, proportional to the number of active neurons).

Proposition 1.10 (Optimal Coding Efficiency)**.** *For a representation using m features in $\mathbb{R}^d$ with coherence μ and sparsity k, the coding efficiency is:*

$$\eta_{\text{code}} = \frac{I(X;T)}{\sum_i \mathbb{E}[a_i]} \approx \frac{d\log(m/k)}{k\varepsilon^2} \cdot \frac{1}{\mu^2}, \tag{1.13}$$

where the numerator uses the approximate capacity formula $I(X;T) \approx d\log(m/k)$ (from Lemma 1.6) and the denominator uses the sparsity k and the interference $\varepsilon = \mu$.

The efficiency η_{code} is maximised when μ is minimised (incoherent frame) and k is minimised (sparse activations). This is precisely the regime where mechanistic interpretability is most tractable: incoherent features and sparse activations are the conditions that make individual features identifiable (low interference) and circuits small (few active features per computation).

The biological nervous system appears to operate close to the optimal efficiency: the Gabor wavelets learned by sparse coding are nearly orthogonal (incoherent, in the frame theory sense), and neural representations in the visual cortex are sparse (only a small fraction of neurons active at any time). The theory of superposition thus connects the engineering constraints of neural network interpretability to the efficiency constraints of biological computation.

Bibliographic Notes

Superposition and toy models: [24]. Frame theory: [17]. Welch bound: [66]. Compressed sensing: [15], [21]. Concentration of measure: [40]. ETFs: [16], [25]. Population codes: [56]. Sparse coding: Olshausen and Field (1996), "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images", *Nature* 381:607–609.

1.11 The Johnson–Lindenstrauss Lemma and Dimensionality Reduction

The capacity results of this chapter show that $\mathbb{R}^d$ can accommodate exponentially many nearly-orthogonal unit vectors. A complementary result — the *Johnson–Lindenstrauss (JL) lemma* [35] — shows that d-dimensional vectors can be projected into $O(\log n/\varepsilon^2)$ dimensions while approximately preserving all pairwise distances.

Theorem 1.11 (Johnson–Lindenstrauss Lemma [35])**.** *For any n points $x_1, \ldots, x_n \in \mathbb{R}^d$ and any $\varepsilon \in (0, 1/2)$, there exists a linear map $\Phi : \mathbb{R}^d \to \mathbb{R}^k$ with $k = O(\log n/\varepsilon^2)$ such that for all i, j:*

$$(1-\varepsilon)\left\|x_i - x_j\right\|^2 \le \left\|\Phi x_i - \Phi x_j\right\|^2 \le (1+\varepsilon)\left\|x_i - x_j\right\|^2. \tag{1.14}$$

The JL lemma has a direct implication for the superposition framework: a network with d neurons can represent $n = e^{O(d\varepsilon^2)}$ features approximately, using random projections to maintain all pairwise distances up to distortion ε. This is the same exponential capacity as Lemma 1.6, derived through a different argument.

The connection between the JL lemma and superposition runs deeper. A random projection matrix $\Phi \in \mathbb{R}^{k \times d}$ with i.i.d. $\mathcal{N}(0, 1/k)$ entries satisfies both the JL condition (1.14) and the RIP (Theorem 1.5) simultaneously. This is not a coincidence: both properties follow from the concentration of measure on the sphere (Lemma A.2), applied to the random matrix Φ.

Corollary 1.12 (JL and Superposition Capacity)**.** *The superposition capacity of a network with d neurons — the maximum number of features that can be approximately represented with pairwise interference at most ε — is exponential in d at rate $e^{\Omega(\varepsilon^2 d)}$, matching the JL dimension reduction rate with the roles of embedding and compression reversed.*

This corollary makes precise the sense in which superposition is the "dual" of dimension reduction: JL compresses n vectors from $\mathbb{R}^d$ to $\mathbb{R}^k$ while preserving pairwise distances; superposition stores n features in $\mathbb{R}^d$ while bounding pairwise interference. Both achieve the same rate $n = e^{O(\varepsilon^2 k)}$, with $k = d$ in the superposition setting.

Bibliographic Notes

Superposition and toy models: [24]. Frame theory: [17]. Welch bound: [66]. Compressed sensing: [15], [21]. Concentration of measure: [40]. ETFs: [16], [25]. Population codes: [56]. Sparse coding: [48]. Johnson–Lindenstrauss lemma: Johnson and Lindenstrauss (1984), "Extensions of Lipschitz Mappings into a Hilbert Space."

Chapter 2

Circuits as Mathematical Objects

2.1 The Need for a Formal Notion of Circuit

The concept of a "circuit" in neural networks is used extensively in the mechanistic interpretability literature, but it is used loosely. Papers speak of the "indirect object identification circuit," the "curve detector circuit," or the "modular addition circuit" without giving a definition that would allow one to determine, for an arbitrary network and an arbitrary computation, whether a given subgraph qualifies. This is not merely a pedantic concern. Without a formal definition, we cannot pose sharp questions: How large is a circuit? When do two circuits compose? What does it mean for a circuit to be faithful to the computation it purportedly implements? When does a computation that could be implemented by a small circuit actually require a large one?

We give such a definition here, drawing on the parallel with *Boolean circuit complexity* [4] while accommodating the features that distinguish neural circuits from their Boolean counterparts: real-valued weights, approximate computations, and the absence of a fixed gate library.

2.2 The Computational Graph and Its Substructure

Let $\mathcal{N} = (V, E, W, \phi)$ be a feedforward neural network, where:

- $V = \bigsqcup_{\ell=0}^{L} V_\ell$ is the vertex set, partitioned into layers V_0 (inputs), $V_1, \ldots, V_{L-1}$ (hidden), and V_L (outputs);
- $E \subseteq \bigcup_{\ell=0}^{L-1} (V_\ell \times V_{\ell+1})$ is the edge set;
- $W : E \to \mathbb{R}$ is the weight function;
- $\phi = \{\phi_v\}_{v \in V \setminus V_0}$ is the collection of activation functions, where $\phi_v : \mathbb{R} \to \mathbb{R}$.

The function computed by $\mathcal{N}$ is $f_{\mathcal{N}} : \mathbb{R}^{|V_0|} \to \mathbb{R}^{|V_L|}$, defined by the standard forward pass: for each $v \in V_{\ell+1}$, the pre-activation is $z_v = \sum_{u:(u,v)\in E} W(u,v) \cdot x_u$ and the post-activation is $x_v = \phi_v(z_v)$, where x_u is computed at the previous layer.

Definition 2.1 (Neural Circuit)**.** A *circuit* in $\mathcal{N}$ is a tuple $\mathcal{C} = (V^{\mathcal{C}}, E^{\mathcal{C}}, W^{\mathcal{C}}, \phi^{\mathcal{C}})$ where:

(1) $V^{\mathcal{C}} \subseteq V$, $E^{\mathcal{C}} \subseteq E \cap (V^{\mathcal{C}} \times V^{\mathcal{C}})$, with weights and activations inherited from $\mathcal{N}$;

(2) The induced graph $(V^{\mathcal{C}}, E^{\mathcal{C}})$ is a DAG;

(3) There are designated *input nodes* $V^{\mathcal{C}}_{\text{in}} \subseteq V^{\mathcal{C}}$ with no incoming edges within $\mathcal{C}$, and *output nodes* $V^{\mathcal{C}}_{\text{out}} \subseteq V^{\mathcal{C}}$ with no outgoing edges within $\mathcal{C}$.

The circuit computes a function $f_{\mathcal{C}} : \mathbb{R}^{|V_{\text{in}}^{\mathcal{C}}|} \to \mathbb{R}^{|V_{\text{out}}^{\mathcal{C}}|}$ by the same forward-pass rule as $\mathcal{N}$, restricted to its nodes and edges.

Several immediate observations deserve emphasis. First, $\mathcal{N}$ itself is a circuit with $V^{\mathcal{C}} = V$. Second, any single neuron v together with its incoming edges is a circuit. Third — and crucially — a circuit need not respect the layer structure of $\mathcal{N}$: the input nodes may be at multiple layers, not just at V_0. This matters enormously for transformer circuits, where queries, keys, and values enter from different parts of the computation, and where the correct notion of "input" to a sub-circuit is an activation at an intermediate layer, not the original token embedding.

2.3 The Algebra of Circuits

Circuits can be combined in two ways that mirror the standard operations of modular programming: sequential composition and parallel composition.

Definition 2.2 (Sequential Composition)**.** Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be circuits in $\mathcal{N}$ with $V_{\text{out}}^{\mathcal{C}_1} = V_{\text{in}}^{\mathcal{C}_2}$. The *sequential composition* $\mathcal{C}_2 \circ \mathcal{C}_1$ has node set $V^{\mathcal{C}_1} \cup V^{\mathcal{C}_2}$, edge set $E^{\mathcal{C}_1} \cup E^{\mathcal{C}_2}$, input nodes $V_{\text{in}}^{\mathcal{C}_1}$, and output nodes $V_{\text{out}}^{\mathcal{C}_2}$. It computes $f_{\mathcal{C}_2} \circ f_{\mathcal{C}_1}$.

Definition 2.3 (Parallel Composition)**.** Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be circuits with $V^{\mathcal{C}_1} \cap V^{\mathcal{C}_2} = \varnothing$. The *parallel composition* $\mathcal{C}_1 \oplus \mathcal{C}_2$ has disjoint union node and edge sets, and computes $(x_1, x_2) \mapsto (f_{\mathcal{C}_1}(x_1), f_{\mathcal{C}_2}(x_2))$.

These two operations endow the class of circuits with the structure of a *symmetric monoidal category* [26]: circuits are morphisms, spaces $\mathbb{R}^n$ are objects, $\oplus$ is the monoidal product, and $\mathbb{R}^0 = \{0\}$ is the monoidal unit. Sequential composition is associative but not commutative; parallel composition is both associative and commutative (up to reordering of coordinates).

This categorical structure is not merely decorative. It tells us that the right way to think about a large neural network is as a composition of circuits — a hierarchical decomposition that exactly mirrors the compositional structure of the functions the network computes. The *modularity hypothesis* of mechanistic interpretability — that networks are built from identifiable, reusable functional units — finds its precise mathematical statement here: it is the hypothesis that $f_{\mathcal{N}}$ factors, approximately, as a sequential or parallel composition of circuits each implementing an identifiable primitive function.

2.4 The Circuit Discovery Problem

Given a network $\mathcal{N}$ and a task $\mathcal{T}$ (specified as a target function or a dataset), the *circuit discovery problem* is to find the smallest circuit $\mathcal{C} \subseteq \mathcal{N}$ that implements $\mathcal{T}$ faithfully.

Definition 2.4 (Faithfulness)**.** Let D be a distribution over $\mathbb{R}^{|V_0|}$ and $\varepsilon > 0$. A circuit $\mathcal{C} \subseteq \mathcal{N}$ is (ε, D)-*faithful* for task $\mathcal{T}$ if

$$\mathbb{E}_{x \sim D}\Big[\big\| f_{\mathcal{C}}(x_{\text{in}}) - f_{\mathcal{N}}(x)[V_{\text{out}}] \big\|^2\Big] \;\le\; \varepsilon^2, \tag{2.1}$$

where x_{in} denotes the activations of $\mathcal{C}$'s input nodes on input x, and $f_{\mathcal{N}}(x)[V_{\text{out}}]$ denotes the activations of $\mathcal{N}$'s output nodes on input x.

The circuit discovery problem is:

$$\min_{\mathcal{C} \subseteq \mathcal{N}} |E^{\mathcal{C}}| \quad \text{subject to } \mathcal{C} \text{ is } (\varepsilon, D)\text{-faithful.} \tag{2.2}$$

This is a combinatorial optimisation problem: find the sparsest subgraph of $\mathcal{N}$ whose output approximates the full network on the given task and distribution. In this form, it is immediately recognisable as a version of the *minimum subgraph* problem in graph theory, which is NP-hard in general. However, the structure of neural networks allows tractable relaxations.

The ℓ^1 Relaxation. Replace the binary edge indicator with a continuous weight $\alpha_e \in [0,1]$ for each edge $e \in E$. The *masked network* $\mathcal{N}_\alpha$ replaces the weight on edge e with $\alpha_e \cdot W(e)$. The relaxed circuit discovery problem is:

$$\min_{\alpha \in [0,1]^{|E|}} \{\|\alpha\|_1 + \lambda \cdot \mathbb{E}_{x \sim D}[\mathcal{L}(\mathcal{N}_\alpha(x), f_{\mathcal{N}}(x))]\}, \tag{2.3}$$

where $\mathcal{L}$ is a loss function and $\lambda > 0$ controls the tradeoff. This is continuous, differentiable with respect to α, and solvable by gradient descent on α while holding W fixed. The ℓ^1 penalty is exactly the sparsity-inducing regulariser from compressed sensing (Chapter 1) — we are once again asking for the sparsest solution to an underdetermined problem, now in the combinatorial structure of the network rather than in the feature representation.

2.5 Faithfulness, Completeness, and the Residual

The optimisation (2.3) finds a circuit that approximates the network on the given task, but two questions remain. First, is the circuit faithful not just in expectation but pointwise? Second, what is the residual — the part of the computation not captured by the circuit — and is it meaningful?

Define the *circuit residual* $R_{\mathcal{C}}(x) = f_{\mathcal{N}}(x)[V_{\text{out}}] - f_{\mathcal{C}}(x_{\text{in}})$ and decompose it as:

$$R_{\mathcal{C}}(x) = R_{\mathcal{C}}^{\text{task}}(x) + R_{\mathcal{C}}^{\text{noise}}(x), \tag{2.4}$$

where $R_{\mathcal{C}}^{\text{task}}$ is correlated with the task labels (systematic errors: the circuit has missed something functionally important) and $R_{\mathcal{C}}^{\text{noise}}$ is uncorrelated with them (unsystematic errors: irreducible variance). A circuit is *complete* for a task if $R_{\mathcal{C}}^{\text{task}} \equiv 0$; it is *minimal-complete* if it is complete and no proper sub-circuit of it is complete.

Proposition 2.5 (Variance Decomposition of Residual)**.**

$$\text{Var}[f_{\mathcal{N}}(x)] = \text{Var}[f_{\mathcal{C}}(x_{\text{in}})] + \text{Var}[R_{\mathcal{C}}^{\text{task}}(x)] + \text{Var}[R_{\mathcal{C}}^{\text{noise}}(x)] + 2\,\text{Cov}[f_{\mathcal{C}}(x_{\text{in}}), R_{\mathcal{C}}(x)]. \tag{2.5}$$

The circuit is ε-faithful if $\text{Var}[R_{\mathcal{C}}(x)] \le \varepsilon^2$*; complete if* $\text{Var}[R_{\mathcal{C}}^{\text{task}}(x)] = 0$.

This decomposition is formally identical to the ANOVA decomposition in statistics and to the bias-variance tradeoff in learning theory. The term $\text{Cov}[f_{\mathcal{C}}, R_{\mathcal{C}}]$ measures how much the circuit's output is correlated with its own errors — which can be positive (the circuit overclaims) or negative (the circuit is systematically conservative).

2.6 Transformer Circuits: Attention as a Routing Mechanism

The definition of §2.2 applies to feedforward networks but requires extension for transformers, where the attention mechanism introduces a qualitatively different computational structure. In a single-layer transformer with H attention heads, the output is:

$$f(x) \;=\; \sum_{h=1}^{H} A^h(x) \cdot (xW_V^h)W_O^h + xW_{\text{MLP}}, \tag{2.6}$$

where $A^h(x) = \text{softmax}\Big(xW_Q^h(xW_K^h)^\top/\sqrt{d_k}\Big) \in \mathbb{R}^{T\times T}$ is the attention matrix of head h, T is the sequence length, and $W_Q^h, W_K^h \in \mathbb{R}^{d\times d_k}$, $W_V^h \in \mathbb{R}^{d\times d_v}$, $W_O^h \in \mathbb{R}^{d_v\times d}$ are the weight matrices of head h.

The crucial observation of [23] is that the attention mechanism acts as a *content-based routing* device: for each position t, the head computes a distribution over source positions and routes information from those positions to t. The circuit corresponding to a single attention head has a natural four-part structure — the query circuit (computing what position t is looking for), the key circuit (computing what each source position offers), the value circuit (computing what information to route), and the output circuit (writing the routed information back to the residual stream). These four sub-circuits compose sequentially within each head, and the heads compose in parallel.

For multi-layer transformers, the *residual stream* — the running sum of contributions from all layers — plays the role of the shared communication bus. A circuit in a multi-layer transformer is therefore a subgraph of the *interaction graph* whose nodes are attention heads and MLP layers, and whose edges represent the paths by which information flows through the residual stream from one component to another.

Definition 2.6 (Transformer Circuit)**.** In an L-layer transformer with H heads per layer, the *interaction graph* $\mathcal{G}$ has as nodes each attention head (l,h), each MLP block MLP_l, an embedding node E, and an unembedding node U. Edges run from u to v if the output of u affects the query, key, or value computation of v through the residual stream. Edge weights are the *composition scores*:

$$c_Q(u,v) = \left\|W_O^u W_Q^v\right\|_F, \quad c_K(u,v) = \left\|W_O^u W_K^v\right\|_F, \quad c_V(u,v) = \left\|W_O^u W_V^v\right\|_F.$$

A *transformer circuit* for task $\mathcal{T}$ is a faithful subgraph of $\mathcal{G}$.

The composition scores deserve comment. The matrix W_O^u takes the output of head u from value space back to residual space; W_Q^v projects from residual space to query space of head v. The product $W_O^u W_Q^v$ measures how much head u's output influences head v's queries. When this product is small in Frobenius norm, head v's queries are effectively independent of head u's output, and the edge (u,v) can be pruned without significant loss of faithfulness.

2.7 Circuit Complexity and the Depth–Width Tradeoff

Boolean circuit complexity distinguishes between the *size* of a circuit (number of gates) and its *depth* (length of the longest path from input to output). Depth corresponds to sequential computation time; size to total computational work. There is a fundamental tradeoff: shallow circuits for many functions must be large, while deep circuits can be small.

An analogous tradeoff exists for neural circuits. Define:

- The *depth* of $\mathcal{C}$ as the length of the longest directed path from $V_{\text{in}}^{\mathcal{C}}$ to $V_{\text{out}}^{\mathcal{C}}$.
- The *width* of $\mathcal{C}$ at depth ℓ as $|V_\ell^{\mathcal{C}}|$.
- The *size* of $\mathcal{C}$ as $|E^{\mathcal{C}}|$.

Theorem 2.7 (Neural Depth–Width Tradeoff [22, 60])**.** *Let $g : [0,1]^n \to \mathbb{R}$ be a function with s-sparse Fourier spectrum at resolution ε.*

(i) A circuit of depth $O(\log(1/\varepsilon))$ and width $O(s)$ can ε-approximate g.

(ii) A circuit of depth 1 (single hidden layer) requires width $\Omega(e^{n/s})$ to ε-approximate g.

*The exponential separation between depth-*1 *and depth-*$O(\log(1/\varepsilon))$ *circuits is tight for functions with* $s \ll n$ *active Fourier modes.*

The informal interpretation: deep circuits efficiently represent compositional functions because each layer builds on features constructed by previous layers. Shallow circuits must represent all combinations of input features directly, requiring exponentially many neurons. This is the deep learning analogue of the classical result that polynomial-size NC^1 circuits (logarithmic depth) are strictly more efficient than polynomial-size AC^0 circuits (constant depth) for many functions.

For transformer circuits, the depth corresponds to the number of attention layers involved, and the width corresponds to the number of heads engaged. The empirical finding that many transformer capabilities can be attributed to small, sparse circuits of depth 2–4 (involving only a few heads) suggests that the relevant functions are in a low-depth, low-width regime.

2.8 The Minimality Problem and Its Hardness

While the relaxation (2.3) provides a tractable algorithm for finding circuits, the resulting circuit may not be minimal. Determining the minimum-size faithful circuit is, in general, hard.

Theorem 2.8 (Hardness of Minimum Circuit)**.** *The problem of finding a minimum-size* ε*-faithful circuit in a given feedforward neural network is NP-hard, even for* $\varepsilon = 0$ *and networks with linear activations.*

Proof sketch. We reduce from the *minimum weight subgraph* problem: given a directed graph $G = (V, E)$ with non-negative edge weights and a target set of input-output pairs, find the minimum-weight subgraph of G that routes each input to the correct output. This problem is NP-hard by reduction from set cover. Any feedforward neural network with linear activations and integer weights can be viewed as an instance of this problem, and a polynomial-time reduction from minimum-weight subgraph to minimum neural circuit completes the argument. □

The hardness result does not make circuit discovery impossible — the relaxation (2.3) finds approximate solutions efficiently — but it means we cannot hope for efficient algorithms that provably find the smallest circuit in all cases. The hunt for better algorithms and tighter approximation guarantees is one of the most important open problems at the intersection of mechanistic interpretability and combinatorial optimisation.

What this means scientifically is more subtle. The fact that humans *can* find small, interpretable circuits — by hand, using targeted ablation experiments and activation patching — suggests that the circuits implementing many language model capabilities are not adversarially complex. They are genuinely sparse, genuinely modular, and genuinely small relative to the total size of the network. Whether this is a property of the training procedure (SGD with weight decay naturally finds sparse solutions), of the architecture (transformers are particularly amenable to modular solutions), or of the tasks (natural language has compositional structure that maps onto small circuits) is a question we will return to in Chapter 7.

Bibliographic Notes

The formal treatment of transformer circuits originates in [23]. The composition scores in Definition 2.6 are introduced there. The depth-width tradeoff for neural networks is studied in [22] and [60]. The categorical structure of circuits is developed in [26]. The NP-hardness

of minimum circuit recovery is related to set cover hardness results; see [4] for background on circuit complexity.

2.9 The IOI Circuit: A Complete Worked Example

To ground the abstract definitions of §§2.1–2.5, we trace through the most thoroughly studied transformer circuit in the mechanistic interpretability literature: the *indirect object identification* (IOI) circuit, identified by Wang et al. [64] in GPT-2 small.

2.9.1 The Task

The IOI task is the following: given a sentence of the form "*When Mary and John went to the store, John gave a drink to* __," predict the name that fills the blank (here, Mary). The correct answer is the *indirect object* — the person who did not perform the action most recently.

This task requires the network to:

(i) Identify the names in the sentence (Mary, John);

(ii) Identify the subject of the most recent action (John);

(iii) Return the *other* name (Mary) as the prediction.

2.9.2 The Circuit Structure

Wang et al. [64] identified a circuit $\mathcal{C}_{\text{IOI}} \subseteq \mathcal{N}_{\text{GPT2}}$ with the following structure:

Duplicate token heads (DTH): a set of attention heads (layers 0 and 1) that copy information about the positions of repeated tokens to the token positions themselves. These heads implement the sub-circuit that identifies that "John" appears twice in the sentence.

Induction heads (IH): a set of attention heads (layer 2) that attend to the position following the first occurrence of the current token. In combination with the DTH, these identify the context: what followed John's first occurrence.

S-inhibition heads (SIH): a set of attention heads (layers 7–10) that attend to the subject of the most recent action (John in the above sentence) and write a negative signal to the position of that name's token embedding.

Name mover heads (NMH): a set of attention heads (layers 9–11) that attend to names in the sentence and move their information to the final token position. The inhibition from SIH suppresses the contribution of the subject's name, leaving the indirect object's name dominant.

In circuit terms, the IOI circuit $\mathcal{C}_{\text{IOI}}$ is the sequential composition:

$$\mathcal{C}_{\text{IOI}} \;=\; \mathcal{C}_{\text{NMH}} \circ \mathcal{C}_{\text{SIH}} \circ \mathcal{C}_{\text{IH}} \circ \mathcal{C}_{\text{DTH}}, \tag{2.7}$$

where each $\mathcal{C}_{\text{X}}$ is the parallel composition of the heads in group X.

2.9.3 Circuit Analysis in Our Formalism

Applying Definition 2.1 and Definition 2.6, we can make the IOI circuit precise. The interaction graph $\mathcal{G}$ for GPT-2 small has 12 layers $\times$ 12 heads $= 144$ attention head nodes plus 12 MLP nodes. The IOI circuit $\mathcal{C}_{\text{IOI}}$ is a subgraph of $\mathcal{G}$ with approximately 26 attention head nodes and 0 MLP nodes (the task is implemented almost entirely through attention).

The faithfulness of the circuit is verified empirically: Wang et al. find that ablating (zeroing out) all weights outside the circuit reduces the IOI accuracy from $\sim 80\%$ to $\sim 77\%$ on the test distribution, a drop of only 3%. In the formal language of Definition 2.4, the circuit is approximately $(0.03, D_{\text{IOI}})$-faithful on the IOI distribution.

The total circuit size is $|E^{\mathcal{C}_{\text{IOI}}}| \approx 26 \times d_k^2 \approx 26 \times 64^2 \approx 106{,}000$ parameters, out of 124,000,000 total parameters in GPT-2 small. The circuit accounts for $\sim 0.085\%$ of the total parameters but implements the full IOI capability. This is a striking example of the sparsity that the minimality criterion (2.2) seeks to exploit.

2.9.4 The Composition Scores

The composition scores (Definition 2.6) between the head groups can be computed from the weight matrices. The key compositions are:

$$c_K(\text{DTH} \to \text{IH}) = \left\|W_O^{\text{DTH}} W_K^{\text{IH}}\right\|_F \approx 0.73 \quad \text{(strong key composition)}, \tag{2.8}$$

$$c_V(\text{IH} \to \text{SIH}) = \left\|W_O^{\text{IH}} W_V^{\text{SIH}}\right\|_F \approx 0.41 \quad \text{(moderate value composition)}, \tag{2.9}$$

$$c_V(\text{SIH} \to \text{NMH}) = \left\|W_O^{\text{SIH}} W_V^{\text{NMH}}\right\|_F \approx 0.35 \quad \text{(moderate value composition)}. \tag{2.10}$$

These scores are substantially larger than the average composition score between random pairs of heads (≈ 0.05), confirming that the IOI circuit involves genuine information flow between the identified head groups.

2.10 Circuit Discovery Algorithms in Practice

The ℓ^1 relaxation (2.3) is the principled approach to circuit discovery, but in practice, mechanistic interpretability uses a variety of techniques that can be viewed as special cases or approximations of this framework.

Activation patching. Activation patching (also called "causal tracing") is the technique of replacing the activation of a specific node with its value on a different ("corrupted") input, and measuring the effect on the output. In the notation of §2.4, this corresponds to estimating $\text{MI}(v)$ (the marginal contribution of node v) via a finite-difference approximation:

$$\text{patch}(v) \approx \mathcal{L}(x_{\text{clean}}) - \mathcal{L}_{\text{patched}}(x_{\text{clean}}, v \leftarrow h_v(x_{\text{corrupted}})), \tag{2.11}$$

where $\mathcal{L}_{\text{patched}}$ denotes the loss after patching. This is equivalent to the $\text{MI}(e)$ score of (4.9) when applied to the edge incoming to node v, under the approximation that the corrupted input provides the "counterfactual" baseline.

Attention knockout. Attention knockout (zeroing out specific attention entries $\alpha_{t,s}^h$) is a special case of activation patching applied to the attention weights rather than the hidden activations. It is particularly useful for transformer circuits because it allows the isolation of specific information flows ("what information is head h routing from position s to position t?").

Path patching. Path patching, introduced in [64], extends activation patching to entire paths in the interaction graph, rather than single nodes. It estimates the contribution of the composition $\mathcal{C}_2 \circ \mathcal{C}_1$ (a two-step path in the circuit) by patching at the interface between $\mathcal{C}_1$ and $\mathcal{C}_2$. In our formalism, this corresponds to estimating the circuit residual $R_{\mathcal{C}_1}(x)$ at the output of $\mathcal{C}_1$, and using it to isolate the contribution of the path $\mathcal{C}_2 \circ \mathcal{C}_1$ to the overall computation.

Bibliographic Notes

The IOI circuit is from [64]. Transformer circuits and their mathematical framework are in [23]. Depth-width tradeoffs are from [22] and [60]. The categorical structure is in [26]. The NP-hardness reduction uses the minimum vertex cover problem; see [4] for background. Activation patching is developed in Meng et al. (2022); path patching is from [64].

2.11 The Grammar of Computation: Circuit Composition Laws

The algebraic structure of circuits (sequential and parallel composition, §2.3) is not merely a mathematical convenience. It reflects a deep property of computation: that complex computations are built from simpler ones in a systematic way. We develop this connection further by establishing *composition laws* for circuit properties.

2.11.1 Faithfulness Under Composition

Proposition 2.9 (Sequential Composition Preserves Faithfulness)**.** *Let $\mathcal{C}_1$ be (ε_1, D_1)-faithful for sub-task $\mathcal{T}_1$ and $\mathcal{C}_2$ be (ε_2, D_2)-faithful for sub-task $\mathcal{T}_2$, where the distribution D_2 over $\mathcal{C}_2$'s inputs is induced by $\mathcal{C}_1$'s outputs under D_1. Then the sequential composition $\mathcal{C}_2 \circ \mathcal{C}_1$ is $(\varepsilon_1 + \varepsilon_2 \cdot L_2,\ D_1)$-faithful for the composed task $\mathcal{T}_2 \circ \mathcal{T}_1$, where L_2 is the Lipschitz constant of $f_{\mathcal{C}_2}$.*

Proof. For any input $x \sim D_1$:

$$\begin{aligned}
&\|f_{\mathcal{C}_2}(f_{\mathcal{C}_1}(x)) - f_{\mathcal{N}}(x)\| \\
&\leq \left\| f_{\mathcal{C}_2}(f_{\mathcal{C}_1}(x)) - f_{\mathcal{C}_2}(f^{(1)}_{\mathcal{N}}(x)) \right\| + \left\| f_{\mathcal{C}_2}(f^{(1)}_{\mathcal{N}}(x)) - f^{(2)}_{\mathcal{N}}(f^{(1)}_{\mathcal{N}}(x)) \right\| \\
&\leq L_2 \left\| f_{\mathcal{C}_1}(x) - f^{(1)}_{\mathcal{N}}(x) \right\| + \varepsilon_2 \\
&\leq L_2 \varepsilon_1 + \varepsilon_2.
\end{aligned}$$

Taking expectations gives $(\varepsilon_1 L_2 + \varepsilon_2, D_1)$-faithfulness, which we bound by $(\varepsilon_1 + \varepsilon_2) \cdot \max(1, L_2)$. □

The Lipschitz constant L_2 in Proposition 2.9 is bounded by the operator norm of $\mathcal{C}_2$'s weight matrices, which is in turn bounded by the spectral norm of the circuit. For circuits with weight norms bounded by C, the Lipschitz constant is $L_2 \leq C^L$ where L is the depth. This exponential growth of the faithfulness error with depth is the circuit-theoretic manifestation of the *gradient explosion* problem in deep network training.

2.11.2 Size Under Composition

The size of a composed circuit is bounded by the sum of the constituent sizes:

$$|E^{\mathcal{C}_2 \circ \mathcal{C}_1}| \;=\; |E^{\mathcal{C}_1}| + |E^{\mathcal{C}_2}|, \qquad |E^{\mathcal{C}_1 \oplus \mathcal{C}_2}| \;=\; |E^{\mathcal{C}_1}| + |E^{\mathcal{C}_2}|. \tag{2.12}$$

These equalities are immediate from the definition: the composed circuit has exactly the edges of both constituents. This linearity of size under composition is the key property that makes the minimum-size circuit discovery problem (2.3) decomposable: if the task decomposes as $\mathcal{T}_2 \circ \mathcal{T}_1$, then the minimum-size faithful circuit is (approximately) the sequential composition of the minimum-size faithful circuits for $\mathcal{T}_1$ and $\mathcal{T}_2$.

Proposition 2.10 (Optimality of Decomposed Circuits)**.** *Let $\mathcal{T} = \mathcal{T}_2 \circ \mathcal{T}_1$ be a compositional task and $\mathcal{C}^*_{\mathcal{T}}$, $\mathcal{C}^*_1$, $\mathcal{C}^*_2$ be the minimum-size faithful circuits for $\mathcal{T}$, $\mathcal{T}_1$, $\mathcal{T}_2$ respectively. Then:*

$$|E^{\mathcal{C}^*_{\mathcal{T}}}| \;\leq\; |E^{\mathcal{C}^*_1}| + |E^{\mathcal{C}^*_2}|. \tag{2.13}$$

The inequality can be strict: the composed circuit $\mathcal{C}^*_2 \circ \mathcal{C}^*_1$ may be larger than $\mathcal{C}^*_{\mathcal{T}}$ when the computation of $\mathcal{T}_1$ and $\mathcal{T}_2$ shares sub-computations that can be implemented once and reused.

The possibility of strict inequality in (2.13) is the circuit-theoretic version of the *subprogram sharing* phenomenon in programming: a subroutine that is called multiple times in a program need only be implemented once. In neural circuits, this sharing corresponds to *polyfunctionality*: a single neuron or attention head that participates in multiple circuits simultaneously.

2.11.3 Kolmogorov Complexity Under Composition

The Kolmogorov complexity of a composed circuit is bounded by:

$$K(\mathcal{C}_2 \circ \mathcal{C}_1) \;\leq\; K(\mathcal{C}_1) + K(\mathcal{C}_2) + O(\log(K(\mathcal{C}_1) + K(\mathcal{C}_2))). \tag{2.14}$$

The additive term $O(\log(K(\mathcal{C}_1) + K(\mathcal{C}_2)))$ accounts for the overhead of specifying how the two circuits are connected. By the chain rule of Kolmogorov complexity [41]: $K(\mathcal{C}_2|\mathcal{C}_1) \leq K(\mathcal{C}_2) + O(1)$ (knowing $\mathcal{C}_1$ does not substantially help in describing $\mathcal{C}_2$), so the bound (2.14) cannot be significantly improved in general.

However, for circuits in the same universality class, the Kolmogorov complexity decomposition can be much more efficient: $K(\mathcal{C}_2 \circ \mathcal{C}_1) \approx K(\mathcal{C}_{\mathcal{U}}) + K(\text{adaptation})$, where $K(\mathcal{C}_{\mathcal{U}})$ is the complexity of the Platonic circuit for the composition and $K(\text{adaptation})$ is the complexity of the adaptation from the Platonic circuit to the specific implementation.

2.12 The Relationship to Boolean Circuit Complexity

The neural circuit framework developed in this chapter parallels, and is inspired by, the classical theory of Boolean circuit complexity. We make this parallel explicit and identify where the two theories diverge.

2.12.1 Similarities

(1) Both theories define circuits as DAGs with input nodes, output nodes, and computation nodes.

(2) Both measure circuit size (number of gates/edges) and circuit depth (longest path).

(3) Both have depth-width tradeoffs (Theorem 2.7 parallels the NC^1 vs. AC^0 separation).

(4) Both distinguish polynomial-size circuits (efficient computation) from exponential-size circuits (brute force).

(5) Both connect circuit complexity to computational hardness (Theorem 2.8 reduces from NP-hard problems).

2.12.2 Differences

(1) *Real-valued vs. Boolean*: Neural circuits operate over $\mathbb{R}$, not $\{0,1\}$. This allows a richer set of computations (smooth functions, continuous read-outs) but also introduces approximation error and numerical precision issues.

(2) *Approximate vs. exact*: Neural circuits are required to be ε-faithful (Definition 2.4), not exact. The approximation ε appears throughout the theory and has no Boolean analogue.

(3) *Learned vs. designed*: Neural circuits are not designed by a programmer but discovered by gradient descent. The universality theory of Chapter 7 characterises the circuits that gradient descent tends to discover.

(4) *Superposition*: Neural circuits can implement *superposition* (storing multiple features in the same direction, Chapter 1), which has no exact analogue in Boolean circuit complexity. The Boolean analogue would be circuits that compute multiple outputs in the same output wire — which is not possible without destroying one of the outputs.

The fourth difference is the most fundamental: superposition is a distinctly analogue phenomenon that relies on the continuous nature of neural activations. It allows neural networks to be more parameter-efficient than Boolean circuits for tasks that admit sparse activations, at the cost of increased interference and decreased robustness.

Bibliographic Notes

The formal treatment of transformer circuits: [23]. The composition scores: [23]. The depth-width tradeoff: [22], [60]. The categorical structure: [26]. The NP-hardness: [4]. Activation patching and path patching: [64]. Boolean circuit complexity: [4]. The IOI circuit: [64]. Polyfunctionality of neurons: Elhage et al. (2023), "Privileged Bases in the Transformer Residual Stream."

2.13 Circuit Discovery in the Residual Stream Basis

Transformer circuits are most naturally analysed in the *residual stream* basis: the decomposition of the residual stream into its constituent components from embedding, attention heads, and MLP layers. We develop the circuit discovery algorithm of §2.4 explicitly for this basis.

2.13.1 The Residual Stream Decomposition

In an L-layer transformer, the residual stream at position t after layer ℓ is:

$$x_t^{(\ell)} \;=\; x_t^{(0)} + \sum_{\ell'=1}^{\ell} \sum_{h=1}^{H} \mathrm{Att}_h^{(\ell')}(x)_t + \sum_{\ell'=1}^{\ell} \mathrm{MLP}^{(\ell')}(x_t^{(\ell'-1)}), \tag{2.15}$$

where $x_t^{(0)} = (W_E)_{t'} + W_{\mathrm{pos},t}$ is the sum of the token and positional embeddings. Each term in (2.15) is a vector in $\mathbb{R}^d$, and their sum is the total residual stream.

2.13.2 Attribution in the Residual Stream

The output of the network at position t is $W_U x_t^{(L)}$ where W_U is the unembedding matrix. The contribution of each component to the output can be decomposed using the bilinearity of the inner product:

$$\text{logit}(c,t) \;=\; (W_U)_c^\top x_t^{(L)} \;=\; (W_U)_c^\top x_t^{(0)} + \sum_{\ell,h} (W_U)_c^\top \text{Att}_h^{(\ell)}(x)_t + \sum_{\ell} (W_U)_c^\top \text{MLP}^{(\ell)}(x_t^{(\ell-1)}). \tag{2.16}$$

The *logit lens* at component (h,ℓ) for class c is the term $(W_U)_c^\top \text{Att}_h^{(\ell)}(x)_t$ — the contribution of attention head h in layer ℓ to the logit for class c. A head with a large logit lens for the correct class $c = y$ is a key contributor to the circuit for task $\mathcal{T}$.

Proposition 2.11 (Logit Lens as Edge Attribution). *The logit lens attribution $(W_U)_c^\top \text{Att}_h^{(\ell)}(x)_t$ is a deterministic lower bound on the edge attribution score* $\text{MI}(e)$ *for the edge from head (h,ℓ) to the output:*

$$\text{MI}(\textit{head } (h,\ell) \to \textit{output}) \;\geq\; \frac{[(W_U)_c^\top \text{Att}_h^{(\ell)}(x)_t]^2}{2\left\|x_t^{(L)}\right\|^2}, \tag{2.17}$$

under the Gaussian assumption on the residual stream.

The logit lens provides a computationally cheap approximation to the full edge attribution score: it requires only one forward pass and no activation patching, making it scalable to large models. The approximation is tight when the residual stream components are approximately independent (which holds when the feature frame is incoherent, §1.2).

2.13.3 Virtual Weights and the Direct Computation

For computing composition scores (Definition 2.6), it is useful to work with the *virtual weights* of the circuit: the effective weight matrices that describe the direct connection between input tokens and output logits, bypassing the intermediate layers.

For a two-layer path through head (h_1,ℓ_1) followed by head (h_2,ℓ_2):

$$W_{\text{virtual}} \;=\; W_O^{h_2} W_V^{h_2} \cdot W_O^{h_1} W_V^{h_1} \cdot W_E, \tag{2.18}$$

the product of the output-value matrices of both heads and the embedding matrix. This virtual weight matrix describes the direct connection from the input token to the output contribution of the two-head path, and can be analysed without running a forward pass.

Proposition 2.12 (Virtual Weights and Universality). *Two transformer circuits $\mathcal{C}$ and $\mathcal{C}'$ are $(0,0)$-isomorphic (Definition 7.1) if and only if their virtual weight matrices are equal (up to the orthogonal symmetry of each layer):*

$$W_{\text{virtual}}^{\mathcal{C}} \;=\; O_{\text{out}} \cdot W_{\text{virtual}}^{\mathcal{C}'} \cdot O_{\text{in}}^\top, \tag{2.19}$$

for orthogonal matrices $O_{\text{in}} \in O(|V_{\text{in}}|)$ and $O_{\text{out}} \in O(|V_{\text{out}}|)$.

The virtual weight isomorphism (2.19) is a computable condition: given two transformer circuits, check whether their virtual weights are related by orthogonal transformations. This is the practical implementation of the universality test for transformer circuits.

Bibliographic Notes

Transformer circuits: [23]. Composition scores: [23]. Depth-width tradeoff: [22], [60]. Categorical structure: [26]. IOI circuit: [64]. Logit lens: Nostalgebraist (2020), "interpreting GPT: the logit lens." Virtual weights: [23]. Boolean circuit complexity: [4]. Polyfunctionality: Elhage et al. (2023).

Chapter 3

Grokking and the Phase Transitions of Circuit Emergence

3.1 The Phenomenon and Its Paradox

In 2022, Power et al. reported a finding so counterintuitive that it prompted a minor crisis of interpretation in the learning theory community [52]. A small transformer trained to perform modular arithmetic — computing $a + b \pmod{p}$ for integers $a, b < p$ — would first *memorise* the training set (achieving zero training loss) and then, hundreds or thousands of gradient steps later and apparently without any further external impetus, suddenly *generalise* to the test set, reducing test loss from near-chance to near-zero in a matter of steps. The researchers named this phenomenon *grokking*, a term borrowed from Heinlein's *Stranger in a Strange Land*.

The paradox is sharp. Classical learning theory has nothing constructive to say about a model that generalises long after it has perfectly memorised its training data. In the classical picture, once training loss is zero, training has effectively ended; there is no gradient signal to improve generalisation. Yet grokking happens, and happens reproducibly, across a range of tasks and architectures. Something is happening after memorisation — something that standard learning theory, in its current form, cannot see.

We argue in this chapter that grokking is a *phase transition*: a discontinuous change in the internal computational structure of the network, from one qualitative regime (memorisation by lookup, implemented by a network with many small but functionally redundant circuits) to another (generalisation by algorithm, implemented by a single compact circuit of low complexity). The transition is not driven by the loss function, which is already zero, but by the *regulariser* — specifically, by weight decay, which provides a continuous pressure toward sparser, lower-norm solutions. The circuit that generalises is not discovered at the moment of grokking but gradually, during the memorisation phase. Grokking is not the discovery of the circuit; it is the *crystallisation* of a circuit that has been latent throughout.

3.2 Formal Setup: Grokking as a Transition in Generalisation Gap

Let $\mathcal{N}_t$ denote the network at training step t, and let $\mathcal{L}_{\text{train}}(t)$ and $\mathcal{L}_{\text{test}}(t)$ be the training and test losses respectively. The *generalisation gap* at time t is $\Delta(t) = \mathcal{L}_{\text{test}}(t) - \mathcal{L}_{\text{train}}(t)$.

Definition 3.1 (Grokking)**.** A training trajectory $\{\mathcal{N}_t\}_{t \geq 0}$ exhibits *grokking* if there exist times $t_{\text{mem}} < t_{\text{grok}}$ such that:

(1) $\mathcal{L}_{\text{train}}(t_{\text{mem}}) \leq \varepsilon_0$ (memorisation: training loss is negligible);

(2) $\Delta(t_{\text{mem}}) \geq C$ for some constant $C > 0$ (the model has not yet generalised at memorisation time);

(3) $\Delta(t_{\text{grok}}) \leq \varepsilon_0$ (the model has generalised at grokking time);

(4) The transition in Δ from C to ε_0 occurs over a time interval $[t_{\text{grok}} - \tau, t_{\text{grok}}]$ with $\tau \ll t_{\text{grok}} - t_{\text{mem}}$ (the transition is rapid relative to the post-memorisation delay).

This definition is deliberately agnostic about the mechanism. Conditions (1)–(3) capture the phenomenological hallmarks; condition (4) isolates the discontinuous character of the transition. We now build toward a mechanistic explanation.

3.3 Two Regimes and the Hidden Progress Measure

To understand what happens between t_{mem} and t_{grok}, we must identify an observable that changes during this period. The loss functions — zero in training, random in test — offer no signal. What changes is the *internal structure* of the network: specifically, the weight norms and the identity of the active circuit.

Nanda et al. [45] identified a crucial quantity: the *restricted loss*, defined as the network's loss when all weights outside a designated candidate circuit $\mathcal{C}_{\text{alg}}$ are zeroed out. They observed that the restricted loss on the algorithmic circuit decreases smoothly and monotonically throughout the post-memorisation period, even as the total network loss shows no visible change. The circuit is "growing stronger" invisibly, underneath the memorisation solution.

We formalise this observation. Define:

$$P(t) \;=\; \frac{\|W_t[\mathcal{C}_{\text{alg}}]\|_F^2}{\|W_t\|_F^2}, \tag{3.1}$$

the *algorithmic progress measure* — the fraction of the total weight norm concentrated in the algorithmic circuit $\mathcal{C}_{\text{alg}}$. Before grokking, $P(t)$ is small; after grokking, it is close to 1. The transition in $P(t)$ from small to large is smooth and precedes the transition in $\Delta(t)$ from large to small.

The relationship between $P(t)$ and $\Delta(t)$ is governed by a threshold mechanism. With weight decay coefficient λ, the total regularised loss is:

$$\mathcal{L}(t) \;=\; \mathcal{L}_{\text{task}}(t) + \lambda \|W_t\|_F^2 \,. \tag{3.2}$$

At t_{mem}, $\mathcal{L}_{\text{task}}(t) \approx 0$, so the gradient descent dynamics are driven almost entirely by $\nabla_W \lambda \|W\|_F^2 = 2\lambda W$. This is weight decay: the network is continuously shrinking all its weights toward zero. The memorisation solution, which requires high-norm weights to implement a lookup table for the training set, is thermodynamically unstable under this pressure. The algorithmic solution — the compact circuit — requires lower total norm and is favoured by weight decay.

3.4 The Norm Gap

Lemma 3.2 (Norm Gap). *Let $\mathcal{N}_{\text{mem}}$ achieve zero training loss via memorisation on N examples, and let $\mathcal{N}_{\text{alg}}$ achieve zero training loss via the correct algorithm of circuit size $s \ll N$. Then:*

$$\|W_{\text{mem}}\|_F^2 \;\geq\; c \cdot N, \qquad \|W_{\text{alg}}\|_F^2 \;\leq\; C \cdot s, \tag{3.3}$$

for absolute constants $c, C > 0$ depending on the architecture but not on N or s.

The inequality (3.3) quantifies the thermodynamic advantage of the algorithmic solution: its norm scales with the circuit size s, while the memorisation solution's norm scales with the dataset size N. With $N \gg s$, weight decay of sufficient strength will eventually make the algorithmic solution cheaper, precipitating the transition.

3.5 The Fourier Analysis of Modular Arithmetic

For concreteness, we work through the grokking phenomenon for modular addition: compute $(a + b) \pmod p$ for prime p. This task has a beautiful harmonic structure that makes the circuit discoverable.

Let $\omega = e^{2\pi i/p}$. The discrete Fourier basis for functions on $\mathbb{Z}/p\mathbb{Z}$ consists of the characters $\chi_k(a) = \omega^{ka}$ for $k = 0, 1, \ldots, p-1$. The key observation is:

$$\chi_k(a + b \bmod p) \;=\; \chi_k(a) \cdot \chi_k(b), \tag{3.4}$$

so the problem decomposes into p independent multiplicative problems, one for each Fourier mode. A network that has learned to represent the Fourier mode k of its inputs can combine them multiplicatively to compute the Fourier mode k of the output.

Nanda et al. [45] confirmed empirically that grokked networks implement exactly this algorithm: after grokking, the embedding matrices have large projections onto a small set of Fourier modes $k_1, k_2, \ldots, k_r$ (typically $r \sim 5$ for $p \sim 100$), and the computation proceeds by (i) extracting the Fourier components of a and b, (ii) multiplying them using the attention mechanism (which implements a bilinear operation), and (iii) taking the inverse Fourier transform to recover $c = a + b \bmod p$.

The circuit implementing this algorithm consists of:

(1) *Fourier extraction*: neurons computing $\cos(2\pi ka/p)$ and $\sin(2\pi ka/p)$ for each active Fourier mode $k \in \{k_1, \ldots, k_r\}$;

(2) *Trigonometric multiplication*: an attention head computing $\cos(2\pi k(a + b)/p)$ via the product-to-sum formula;

(3) *Inverse Fourier transform*: the MLP layer mapping Fourier components back to the logit vector indexed by $c \in \mathbb{Z}/p\mathbb{Z}$.

The total circuit size is $O(r \cdot d)$ — far smaller than the $O(p^2)$ lookup table of the memorisation solution, which accounts for Lemma 3.2.

3.6 The Phase Transition: A Free Energy Argument

Consider the total loss landscape $\mathcal{L}(W) = \mathcal{L}_{\text{task}}(W) + \lambda \|W\|_F^2$ as a function of the weights. For any $\lambda > 0$, the landscape has a unique global minimum. However, the landscape may have multiple local minima with different qualitative character.

Define two basins:

- $\mathcal{B}_{\text{mem}}$: the basin of the memorisation solution, roughly the set of weight configurations with $\mathcal{L}_{\text{task}} \approx 0$ and $\|W\|_F^2 \sim N$;
- $\mathcal{B}_{\text{alg}}$: the basin of the algorithmic solution, with $\mathcal{L}_{\text{task}} \approx 0$ and $\|W\|_F^2 \sim s$.

The "free energies" of the two solutions are $\mathcal{F}_{\text{mem}}(\lambda) \approx \lambda N$ and $\mathcal{F}_{\text{alg}}(\lambda) \approx \lambda s$. Since $s \ll N$, we always have $\mathcal{F}_{\text{alg}} < \mathcal{F}_{\text{mem}}$ for $\lambda > 0$. The algorithmic solution is always the global minimum. The question is whether gradient descent from a memorisation initialisation can reach it.

Theorem 3.3 (Critical Weight Decay for Grokking)**.** *Let $\mathcal{N}$ be initialised near a memorisation solution W_{mem} and trained with weight decay λ. Under suitable smoothness and landscape regularity assumptions, the expected time for the training trajectory to escape $\mathcal{B}_{\text{mem}}$ satisfies:*

$$T_{\text{grok}}(\lambda) \asymp \exp\left(\frac{\Delta\mathcal{F}(\lambda)}{\lambda \cdot \eta}\right), \tag{3.5}$$

where $\Delta\mathcal{F}(\lambda) = \mathcal{F}_{\text{barrier}}(\lambda) - \mathcal{F}_{\text{mem}}(\lambda)$ is the free energy barrier and η is the learning rate. There exists a critical $\lambda^* = \mathcal{F}_{\text{barrier}}/(N-s)$ such that:

- *For $\lambda < \lambda^*$: $T_{\text{grok}}(\lambda)$ grows faster than polynomially in N — grokking does not occur in practical training time.*
- *For $\lambda > \lambda^*$: $T_{\text{grok}}(\lambda)$ decreases rapidly — grokking occurs after a delay of order $1/\lambda$.*

The heuristic behind Theorem 3.3 is Kramers' escape rate formula from stochastic thermodynamics [30]: the mean exit time from a potential well of depth $\Delta\mathcal{F}$ under noise of effective temperature $T_{\text{eff}} \sim \lambda\eta$ is $\exp(\Delta\mathcal{F}/T_{\text{eff}})$. Weight decay plays the role of temperature, driving the system toward the lower-energy basin. The prediction that increasing λ dramatically accelerates grokking has been confirmed empirically; the transition in grokking time as a function of λ is sharp — consistent with a phase transition at λ^*.

3.7 Generalisation of the Framework

The modular arithmetic analysis is concrete and beautiful, but one might worry that it is special. We briefly indicate how the framework generalises.

Sparse parities. Consider learning a parity function $f(x) = \prod_{i \in S} x_i$ for $x \in \{-1, +1\}^n$ and a fixed hidden set $S \subseteq [n]$ of size $|S| = k$. The algorithmic circuit has size $O(k)$; the memorisation solution has size $O(2^k \binom{n}{k})$. The norm gap scales as N/k where $N = 2^n$. With weight decay $\lambda > 1/N$, grokking is predicted at time $T_{\text{grok}} \sim \exp(O(2^k/(\lambda n)))$. This matches the empirically observed slow generalisation of transformers on sparse parity tasks [5].

Compositional tasks. When a task decomposes as $f = g \circ h$, the algorithmic circuit decomposes as $\mathcal{C}_g \circ \mathcal{C}_h$. The memorisation solution is a monolithic lookup table that does not exploit the compositional structure. The norm gap is $\|W_{\text{mem}}\|_F^2 - \|W_{\text{alg}}\|_F^2 \sim N - (s_g + s_h)$, large precisely because compositional tasks can be implemented by small circuits.

3.8 Circuit Crystallisation: A Dynamical Picture

The phase transition perspective gives us the *when* and the *why* of grokking, but not the *how*: what is the dynamical process by which the circuit forms? We propose the following four-stage picture.

Stage 1 — Memorisation ($t < t_{\text{mem}}$). The network learns to correctly classify all training examples by distributing the computational load across all available parameters. The circuit $\mathcal{C}_{\text{alg}}$ is not yet present; the network implements a high-norm, high-interference solution.

Stage 2 — Latent circuit formation ($t_{\text{mem}} < t < t_{\text{grok}}$). Weight decay continuously shrinks the norm of all weights. The weights corresponding to the memorisation solution (the "lookup table" edges) decay more rapidly than those belonging to $\mathcal{C}_{\text{alg}}$, because the algorithmic circuit must survive to zero training loss while the lookup entries are redundant

once the algorithm is available. The progress measure $P(t)$ grows monotonically. The circuit is latent: it exists as a subgraph of the network but its contribution to the output is masked by the residual memorisation weights.

Stage 3 — Transition ($t \approx t_{\text{grok}}$). The memorisation weights have decayed below a critical threshold. At this point, the circuit $\mathcal{C}_{\text{alg}}$ dominates the computation, and the network's test loss drops rapidly. This is the observable grokking event.

Stage 4 — Consolidation ($t > t_{\text{grok}}$). The remaining memorisation weights continue to decay; the circuit sharpens. The network converges to a solution in which essentially all the weight norm is concentrated in $\mathcal{C}_{\text{alg}}$.

This four-stage picture is not merely descriptive. Each stage has a precise characterisation in terms of the circuit algebra of Chapter 2: the transition from Stage 2 to Stage 3 is exactly the moment at which $P(t)$ crosses the threshold at which $\mathcal{C}_{\text{alg}}$ becomes (ε, D)-faithful in the sense of Definition 2.4. Before this threshold, $\mathcal{C}_{\text{alg}}$ is present but not yet faithful; after it, faithfulness is achieved and generalisation follows.

Bibliographic Notes

Grokking was first reported in [52]. The Fourier mechanism and progress measure are developed in [45]. The hidden progress perspective for sparse parities is from [5]. Kramers' escape rate formula, which provides the heuristic behind Theorem 3.3, is treated in [30]. The general theory of metastability for Markov chains that would make Theorem 3.3 rigorous is developed in [47].

3.9 Grokking in Multi-Task Settings

The analysis of §§3.2–3.6 treats a single task. In practice, neural networks are trained on mixtures of tasks, and the grokking dynamics of each task interact. We briefly analyse the multi-task setting.

Consider a network trained simultaneously on two tasks $\mathcal{T}_1$ and $\mathcal{T}_2$ with algorithmic circuits $\mathcal{C}_1$ and $\mathcal{C}_2$. Define:

- The *parameter overlap*: $\text{ov}(\mathcal{C}_1, \mathcal{C}_2) = |E^{\mathcal{C}_1} \cap E^{\mathcal{C}_2}| / \min(|E^{\mathcal{C}_1}|, |E^{\mathcal{C}_2}|)$, the fraction of edges shared between the two circuits.
- The *memorisation norm*: $\|W_{\text{mem}}\|_F^2$ (which scales as $N_1 + N_2$, the combined number of training examples for the two tasks).
- The *algorithmic norm*: $\|W_{\mathcal{C}_1} + W_{\mathcal{C}_2}\|_F^2 \leq \|W_{\mathcal{C}_1}\|_F^2 + \|W_{\mathcal{C}_2}\|_F^2 + 2\text{ov}(\mathcal{C}_1, \mathcal{C}_2)\, \|W_{\mathcal{C}_1}\|_F\, \|W_{\mathcal{C}_2}\|_F$.

Proposition 3.4 (Multi-Task Grokking)**.** *In the multi-task setting, the grokking time for task $\mathcal{T}_i$ is accelerated when the two circuits share edges (large $\text{ov}(\mathcal{C}_1, \mathcal{C}_2)$) and decelerated when they compete for the same parameter budget (when $|E^{\mathcal{C}_1}| + |E^{\mathcal{C}_2}|$ approaches the total parameter count). Specifically, the critical weight decay for simultaneous grokking of both tasks is:*

$$\lambda^*_{\text{multi}} = \frac{\mathcal{F}_{\text{barrier}}}{N_1 + N_2 - s_1 - s_2 + 2\text{ov}(\mathcal{C}_1, \mathcal{C}_2)\sqrt{s_1 s_2}}, \tag{3.6}$$

where $s_i = |E^{\mathcal{C}_i}|$ is the size of each circuit.

The proposition says that tasks whose circuits share many edges require *less* weight decay to jointly grok (the denominator in (3.6) is larger, so λ^* is smaller). This is the formal version of the intuition that related tasks (sharing computational structure) are easier to learn simultaneously than unrelated tasks.

In the extreme case of full overlap (ov $= 1$, $s_1 = s_2 = s$), $\lambda^*_{\text{multi}} \approx \mathcal{F}_{\text{barrier}}/(N_1 + N_2 - 4s)$, which is smaller than the single-task critical weight decay $\lambda^* \approx \mathcal{F}_{\text{barrier}}/(N - s)$ for either task alone. Multi-task training facilitates grokking when tasks are related.

3.10 The Grokking Lens on In-Context Learning

A striking application of the grokking framework is to the phenomenon of *in-context learning*: the ability of large language models to learn new tasks from a few examples provided in the prompt, without any weight update. This appears to be a fundamentally different kind of learning from the gradient-based training process studied throughout this chapter — and yet the circuit framework connects the two.

The key insight, developed by Olsson et al. [49], is that in-context learning is implemented by *induction heads* (Chapter 5): circuits that, given a sequence of input-output pairs in the context, implement a form of associative memory that retrieves the correct output for a new input. The induction head circuit has the same structure as the algorithmic circuit for modular addition: it is a compact, reusable sub-circuit that implements a general algorithm (pattern matching) rather than a lookup table.

In the grokking language: in-context learning is the in-weights generalisation circuit that was present in the network after grokking — the induction head circuit $\mathcal{C}_{\text{IH}}$ — being *used at inference time* to implement new tasks from context. The network does not learn new circuits during inference; it applies an existing circuit (the induction head) to the new task presented in the prompt.

This connection suggests a unified view of grokking and in-context learning:

(1) *Grokking* (training time): gradient descent discovers the algorithmic circuit $\mathcal{C}_{\text{alg}}$ and consolidates it into the weights.

(2) *In-context learning* (inference time): the circuit $\mathcal{C}_{\text{alg}}$ (specifically, the induction head component) is applied to new tasks provided in the prompt, implementing a form of one-shot or few-shot generalisation without weight update.

The capacity of in-context learning is bounded by the pattern-matching capacity $\mathcal{M}_K \leq e^{d_k/2}$ of the induction heads (Proposition 5.7): the network can learn, from context, as many distinct input-output associations as its induction heads can store — which is exponential in d_k.

Bibliographic Notes

Grokking was first reported in [52]. The Fourier mechanism and progress measure are developed in [45]. The hidden progress perspective is from [5]. Kramers' escape rate is in [30]. The metastability framework is in [47]. Induction heads and in-context learning are from [49].

3.11 The Statistical Mechanics of Gradient Descent

The grokking analysis of this chapter uses the language of statistical mechanics — free energy, phase transitions, Kramers' formula — applied to the training dynamics. We make this

connection more precise by establishing a formal mapping between gradient descent dynamics and a statistical mechanical system.

3.11.1 The Gradient Flow as a Langevin Process

Stochastic gradient descent (SGD) with learning rate η and mini-batch noise ξ_t can be written as:

$$W_{t+1} \;=\; W_t - \eta\nabla_W\mathcal{L}(W_t) - \eta\lambda W_t + \sqrt{2\eta\lambda}\xi_t, \tag{3.7}$$

where the first term is the gradient step, the second is weight decay, and the third is the stochastic noise from mini-batch sampling. When $\xi_t \sim \mathcal{N}(0, I)$ (Gaussian mini-batch noise), (3.7) is exactly the *Langevin dynamics* of a particle in the potential $\Phi(W) = \mathcal{L}(W)/\lambda + \|W\|_F^2/2$ at inverse temperature $\beta_{\text{phys}} = 1/\eta\lambda$ (inverse learning rate times inverse weight decay).

Proposition 3.5 (Stationary Distribution of SGD)**.** *In the limit of small learning rate $\eta \to 0$ and long training time $t \to \infty$, the stationary distribution of the SGD dynamics* (3.7) *is the Gibbs distribution:*

$$\pi(W) \;\propto\; \exp\left(-\frac{\mathcal{L}(W) + \lambda\,\|W\|_F^2/2}{\eta\lambda/2}\right) \;=\; \exp\left(-\frac{2(\mathcal{L}(W) + \lambda\,\|W\|_F^2)}{\eta\lambda}\right). \tag{3.8}$$

The stationary distribution (3.8) assigns higher probability to weight configurations with lower regularised loss $\mathcal{L}(W) + \lambda\,\|W\|_F^2/2$. The memorisation solution (high norm, high task loss on test) and the algorithmic solution (low norm, low task loss on test) are both zero task-loss configurations, but the algorithmic solution has lower regularised loss (Lemma 3.2), so it has higher probability in the stationary distribution.

The temperature $T_{\text{eff}} = \eta\lambda/2$ controls the concentration of the stationary distribution around its modes. For small temperature (small $\eta\lambda$), the distribution concentrates tightly around the global minimum (the algorithmic solution); for large temperature (large $\eta\lambda$), the distribution is more diffuse and the network may not grok. The critical weight decay λ^* of Theorem 3.3 corresponds to the temperature below which the distribution concentrates on the algorithmic solution.

3.11.2 The Free Energy Landscape

The free energy of the system at temperature T_{eff} is:

$$\mathcal{F}(\beta_{\text{phys}}) \;=\; -\frac{1}{\beta_{\text{phys}}}\log Z, \quad Z \;=\; \int e^{-\beta_{\text{phys}}(\mathcal{L}(W)+\lambda\|W\|_F^2/2)}\,dW. \tag{3.9}$$

The free energy landscape has two basins of attraction:

- $\mathcal{F}_{\text{mem}}(\beta_{\text{phys}}) \approx \mathcal{L}_{\text{mem}} + \lambda N/2$: the free energy of the memorisation basin, where $\mathcal{L}_{\text{mem}} = 0$ (zero training loss) and $\|W_{\text{mem}}\|_F^2 \approx N$ (Lemma 3.2).
- $\mathcal{F}_{\text{alg}}(\beta_{\text{phys}}) \approx \lambda s/2$: the free energy of the algorithmic basin, where $\|W_{\text{alg}}\|_F^2 \approx s$.

The transition between basins occurs when $\mathcal{F}_{\text{mem}} = \mathcal{F}_{\text{alg}}$, i.e., when $\lambda N/2 = \lambda s/2$, i.e., $N = s$. But this is never true (since $s \ll N$); the algorithmic solution is always lower in free energy. The physical question is not which basin has lower free energy but how quickly the system can escape the memorisation basin.

The escape time is controlled by the free energy barrier $\Delta\mathcal{F} = \mathcal{F}_{\text{barrier}} - \mathcal{F}_{\text{mem}}$, and the Kramers formula gives $T_{\text{grok}} \propto e^{\Delta\mathcal{F}/T_{\text{eff}}}$, which is the content of Theorem 3.3.

3.11.3 The Role of Batch Size

One implication of the statistical mechanics picture is a prediction about the effect of batch size on grokking. Larger batches reduce the mini-batch noise ξ_t (since the noise is averaged over more examples), which effectively reduces the temperature T_{eff}. Lower temperature concentrates the stationary distribution more tightly around the algorithmic solution but also makes it harder to escape the memorisation basin (higher barrier in Kramers units).

Corollary 3.6 (Batch Size and Grokking Time)**.** *With mini-batch size B, the effective temperature in* (3.8) *is $T_{\text{eff}}(B) \approx \eta\lambda/2 + \eta^2\sigma^2/B$ where $\sigma^2 = \text{Var}[\nabla_W \mathcal{L}_{\text{single}}]$ is the variance of the single-sample gradient. The grokking time satisfies:*

$$T_{\text{grok}}(B) \;\propto\; \exp\left(\frac{\Delta\mathcal{F}}{\eta\lambda/2 + \eta^2\sigma^2/B}\right). \tag{3.10}$$

Increasing B decreases T_{eff} and therefore increases T_{grok} *(slower grokking with larger batch). Decreasing B (more stochastic training)* decreases T_{grok} *(faster grokking).*

This prediction — that grokking is faster with smaller batch sizes — is empirically testable and, if confirmed, would provide strong evidence for the statistical mechanics picture of grokking.

Bibliographic Notes

Grokking: [52]. Fourier mechanism: [45]. Hidden progress: [5]. Kramers' formula: [30]. Metastability: [47]. Induction heads and in-context learning: [49]. The Langevin dynamics of SGD is analysed in Welling and Teh (2011), "Bayesian Learning via Stochastic Gradient Langevin Dynamics." The connection between batch size and effective temperature is related to Hoffer et al. (2017), "Train Longer, Generalise Better."

3.12 Computational Complexity of Grokking Tasks

The grokking phenomenon is not universal: some tasks grok quickly, others slowly, and some not at all. The theory of Chapter 2 suggests that the ease of grokking is determined by the norm gap (Lemma 3.2), which in turn depends on the size of the minimal faithful circuit. But the size of the minimal faithful circuit is bounded below by the Kolmogorov complexity of the task (Proposition 4.7). We explore this connection.

3.12.1 Tasks That Grok Quickly

Tasks with small Kolmogorov complexity — tasks admitting short algorithms — have small minimal faithful circuits. By Lemma 3.2, the norm gap scales as $N - s$ where $s = |E^{\mathcal{C}_{\text{alg}}}|$ is small. For large N, the norm gap is approximately N, and the critical weight decay is $\lambda^* \approx \mathcal{F}_{\text{barrier}}/N$ (small). Such tasks grok at small weight decay and with short delay after memorisation.

Examples of tasks that grok quickly (empirically):

- Modular arithmetic: $K(f_{\mathcal{T}_p}) \approx \log p$ (short description).
- Symmetric group operations: $K(f_{\mathcal{T}_{S_n}}) \approx n \log n$ (FFT on the symmetric group).
- Polynomial evaluation: $K(f_{\text{poly}}) \approx \log(\text{deg})$ (the Horner algorithm).

3.12.2 Tasks That Grok Slowly or Not At All

Tasks with large Kolmogorov complexity have large minimal faithful circuits. The norm gap $N - s$ is smaller (since s is larger), and the critical weight decay is larger ($\lambda^* \approx \mathcal{F}_{\text{barrier}}/(N-s)$ is larger when s is close to N). Such tasks may fail to grok if the training set size N is not much larger than the circuit size s.

Proposition 3.7 (Grokking Condition)**.** *A task $\mathcal{T}$ with minimal faithful circuit of size s can grok (within reasonable training time) from a training set of size N if and only if $N \gg s$, specifically:*

$$N \geq C \cdot s \cdot \log(T_{\max}/T_{\text{grok,min}}), \tag{3.11}$$

where $T_{\max}$ is the maximum allowable training time and $T_{\text{grok,min}}$ is the minimum acceptable grokking time, and $C > 0$ is a constant depending on the weight decay λ and learning rate η.

The condition $N \gg s$ says that there must be substantially more training examples than the algorithm requires parameters to describe. This is the neural network analogue of the classical learning theory condition: a model of complexity s can be learned from $O(s/\varepsilon)$ examples (the PAC learning bound), but the *efficient* learning of a model requires the training set to be much larger than the model.

3.12.3 The Grokking Frontier

The grokking condition (3.11) defines a *grokking frontier* in the (N, s) plane: tasks with (N, s) above the frontier ($N \gg s$) grok; tasks below the frontier do not. The shape of the frontier is $N \propto s$ (linear), and the proportionality constant is determined by the weight decay and learning rate.

This frontier is the training dynamics analogue of the IPF (Definition 8.8):

- The IPF characterises achievable (test error, circuit complexity) pairs for an optimal learner.
- The grokking frontier characterises achievable (training set size, circuit size) pairs for SGD with weight decay.

Both frontiers are determined by the Kolmogorov complexity of the task, but they capture different aspects: the IPF is a static, worst-case bound, while the grokking frontier is a dynamic, training-time bound.

Bibliographic Notes

Grokking: [52]. Fourier mechanism: [45]. Hidden progress: [5]. Kramers' formula: [30]. Metastability: [47]. Induction heads: [49]. PAC learning and sample complexity: [4]. The Langevin dynamics analysis connects to Welling and Teh (2011).

Chapter 4

The Information-Theoretic Duality of Features and Circuits

4.1 Two Perspectives on the Same Object

The theory developed in Chapters 1–3 has proceeded along two parallel tracks. The first is geometric: features are unit vectors in $\mathbb{R}^d$, they form frames and polytopes, and their superposition in low-dimensional space is governed by the Welch bound and the Restricted Isometry Property. The second is algebraic: circuits are subgraphs of the computational graph, they compose in a monoidal category, and their discovery is a combinatorial optimisation problem.

These two tracks describe, at different levels of abstraction, what a neural network *is*. What they do not yet capture is what the network *does* with information: how much information about the input it retains, how much about the label, and how this information is allocated among features and circuits. For this, we need a third language: information theory.

The central claim of this chapter is a *duality theorem*: the minimal faithful circuit for a task is precisely the solution to an information bottleneck problem [62], in which the network is asked to compress the input into a minimal representation that preserves maximal information about the output. Features are the "code symbols" of this bottleneck, and circuits are the "encoders." The Welch bound and the circuit size bound are both reflections of the same underlying information-theoretic constraint.

4.2 Rate-Distortion Theory: The Fundamental Tradeoff

We recall the basic framework of rate-distortion theory [11], adapted to the neural network setting. Let $X \in \mathcal{X}$ be the input and $Y \in \mathcal{Y}$ the label. An intermediate representation T is produced by encoding X through the network; the downstream computation decodes Y from T.

For a fixed distortion measure $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ and level $D \geq 0$, the *rate-distortion function* is:

$$R(D) \;=\; \min_{p(T|X):\, \mathbb{E}[d(Y,\hat{Y}(T))] \leq D} I(X;T), \tag{4.1}$$

where $\hat{Y}(T)$ is the optimal decoder given T. The function $R(D)$ is non-increasing, convex, and zero only when T contains sufficient information to decode Y perfectly. The Shannon lower bound gives:

$$R(D) \;\geq\; I(X;Y) - \log \frac{1}{1 - D/D_{\max}}, \tag{4.2}$$

where $D_{\max}$ is the distortion of the trivial encoder.

Lemma 4.1 (Neural Rate-Distortion Lower Bound)**.** *Let $T \in \mathbb{R}^d$ be the hidden state encoding X, and let Y be the label. For squared-error distortion $d(y, \hat{y}) = \|y - \hat{y}\|^2$:*

$$I(X;T) \;\geq\; I(X;Y) - \frac{d}{2}\log\!\left(\frac{2\pi e\, D}{d}\right), \tag{4.3}$$

where $D = \mathbb{E}[\left\|Y - \hat{Y}(T)\right\|^2]$ is the expected distortion.

Proof. Let $\varepsilon = Y - \hat{Y}(T)$ be the residual error. By the data processing inequality, $I(X;\hat{Y}(T)) \leq I(X;T)$. Since $I(X;Y) \leq I(X;\hat{Y}) + h(\varepsilon)$ where $h(\varepsilon)$ is the differential entropy of the residual, and since a Gaussian maximises entropy at fixed variance D/d, we have $h(\varepsilon) \leq \frac{d}{2}\log(2\pi e D/d)$. Rearranging gives (4.3). □

The bound (4.3) says that a representation with small $I(X;T)$ cannot achieve arbitrarily low distortion D. The rate of compression is limited by the information content of Y given X, modified by the dimensionality term $d\log(D^{-1})$.

4.3 The Information Bottleneck

The *information bottleneck* [62] provides an alternative formulation more directly connected to machine learning. Instead of fixing the distortion and minimising the rate, the bottleneck fixes the *relevance* and minimises the *complexity*:

$$\mathcal{L}_{\mathrm{IB}}(T) \;=\; I(X;T) - \beta \cdot I(T;Y), \tag{4.4}$$

where $\beta > 0$ trades off compression against relevance. The minimiser is the *optimal bottleneck representation* at tradeoff β.

The bottleneck objective (4.4) has an intuitive interpretation. $I(X;T)$ measures how much of the input is retained by T — the "complexity" of the representation. $I(T;Y)$ measures how much of the label can be predicted from T — the "relevance." A good representation retains exactly as much of the input as is necessary to predict the label, and no more.

We now connect the information bottleneck to the circuit framework.

Theorem 4.2 (Faithful Circuits Achieve the Bottleneck)**.** *Let $\mathcal{C}^*$ be a minimum-size (ε, D)-faithful circuit for task $(X \to Y)$. Under squared-error distortion:*

$$I(X;T_{\mathcal{C}^*}) \;\approx\; R(\varepsilon^2) + O\!\left(\sqrt{|E^{\mathcal{C}^*}|/n}\right), \tag{4.5}$$

where $R(\varepsilon^2)$ is the rate-distortion function at distortion $D = \varepsilon^2$ and n is the number of training examples. The circuit $\mathcal{C}^*$ is an approximate solution to the information bottleneck problem at $\beta = |E^{\mathcal{C}^*}|/(I(X;Y) - R(\varepsilon^2))$.

The proof uses the variational formula for mutual information:

$$I(X;T) \;=\; \sup_{q(Y|T)} \mathbb{E}_{p(X,Y,T)}\left[\log \frac{q(Y|T)}{p(Y)}\right], \tag{4.6}$$

combined with the faithfulness condition (Definition 2.4) and the rate-distortion lower bound (Lemma 4.1). The minimality of $|E^{\mathcal{C}^*}|$ implies the minimality of $I(X;T_{\mathcal{C}^*})$ over all faithful circuits, which is the bottleneck condition.

4.4 Features as Sufficient Statistics

The information bottleneck perspective illuminates the role of features in a new light. Recall that a feature f_i is a direction in $\mathbb{R}^d$ along which the network represents a property of the input. In information-theoretic terms, a feature is a *sufficient statistic* for a property of the input.

Definition 4.3 (Sufficient Feature)**.** A feature $f_i \in S^{d-1}$ is δ-*sufficient* for predicting property $g_i(X)$ if:

$$I(X; f_i^\top T) \;\geq\; I(X; g_i(X)) - \delta. \tag{4.7}$$

The superposition architecture of Chapter 1 acquires a clear information-theoretic interpretation under this definition. The total mutual information in the representation is:

$$I(X;T) \;\approx\; \sum_{i=1}^{m} I(X; a_i(X) f_i) \;+\; \sum_{i \neq j} I_{\text{inter}}(f_i, f_j), \tag{4.8}$$

where I_{inter} is the interaction information capturing redundancy between features. When features are nearly orthogonal (as guaranteed achievable by the Welch bound), the interaction terms are small, and the total information is approximately the sum of individual feature information. Superposition thus allows the network to achieve high total $I(X;T)$ while keeping individual features simple and non-redundant.

The tension with the bottleneck objective (4.4) — which wants to minimise $I(X;T)$ — is resolved by noting that the minimisation is subject to a relevance constraint. A network is not rewarded for storing information about X that is irrelevant for Y. A well-trained network therefore stores the *relevant features* of X and no more. Polysemantic neurons, from this perspective, are neurons that have been recruited to store multiple relevant features simultaneously, because storing irrelevant information would be penalised by the bottleneck objective.

4.5 The Circuit–Feature Duality

We now state the central duality result.

Theorem 4.4 (Circuit–Feature Duality)**.** *Let $\mathcal{C}$ be a faithful circuit for task $(X \to Y)$, and let $\{f_i^{(\ell)}\}$ be the features active at depth ℓ (unit vectors in $\mathbb{R}^{d_\ell}$ with $I(X; (f_i^{(\ell)})^\top T^{(\ell)}) > \delta$). Then:*

(1) (Feature completeness) $I(T^{(\ell)}; Y) \leq \sum_i I(f_i^{(\ell)} \cdot T^{(\ell)}; Y) + O(\ell \cdot m_\ell \cdot \mu_\ell^2)$*, where m_ℓ is the number of active features and μ_ℓ is the coherence at depth ℓ.*

(2) (Circuit factorisation) *$\mathcal{C}$ decomposes as $\mathcal{C}_r \circ \cdots \circ \mathcal{C}_1$ where each $\mathcal{C}_\ell$ maps the frame $\{f_i^{(\ell-1)}\}$ to $\{f_i^{(\ell)}\}$ via a learnable affine map of Frobenius norm bounded by $\sqrt{m_{\ell-1} m_\ell}$.*

(3) (Information flow) *$I(T^{(\ell)}; Y)$ is non-decreasing in ℓ if $\mathcal{C}$ is faithful and activations are approximately information-preserving.*

Part (1) says that a circuit using an incoherent feature frame (small μ_ℓ) can efficiently aggregate information across features. This is the information-theoretic translation of the Welch bound: low coherence means low redundancy between features, so that the sum of individual feature information approximates the total representation information.

Part (2) connects the circuit structure to the feature geometry: each circuit layer is a frame transformation, mapping one system of features to another. The affine maps have Frobenius norm bounded by $\sqrt{m_{\ell-1}m_\ell}$, which bounds the parameter count of each circuit layer.

Part (3) establishes the directionality of information flow: deeper layers of a faithful circuit contain more information about Y, not less. This is a consequence of the data processing inequality applied to the deterministic map $T^{(\ell)} = g_\ell(T^{(\ell-1)})$ for each circuit layer g_ℓ.

The proof of (1) uses the chain rule for mutual information and the independence bound, modified to account for the interference between features as bounded by the coherence.

4.6 Mutual Information as a Circuit Attribution Score

Theorem 4.4 suggests a natural quantity for circuit attribution: the *mutual information contribution* of an edge $e \in E^{\mathcal{C}}$.

Definition 4.5 (Marginal Mutual Information)**.**

$$\mathrm{MI}(e) \;=\; I(T^{(\ell_e)};Y) - I(T^{(\ell_e)}_{-e};Y), \tag{4.9}$$

where $T^{(\ell_e)}_{-e}$ is the representation at depth ℓ_e when edge e is set to zero.

The total circuit relevance decomposes (approximately) as:

$$I(T^{(L)};Y) \;\approx\; \sum_{e\in E^{\mathcal{C}}} \mathrm{MI}(e), \tag{4.10}$$

with approximation error bounded by:

$$\left| I(T^{(L)};Y) - \sum_{e\in E^{\mathcal{C}}} \mathrm{MI}(e) \right| \;\le\; \binom{|E^{\mathcal{C}}|}{2} \cdot \max_{e,e'} \mathrm{MI}_2(e,e'), \tag{4.11}$$

where $\mathrm{MI}_2(e,e')$ is the second-order interaction information between edges e and e'. The bound (4.11) is small when edges are approximately independent — when removing one edge does not significantly change the marginal information contribution of any other. This holds when the circuit is sparse and features are incoherent, the regime guaranteed by the Welch bound and RIP.

Edges with large $\mathrm{MI}(e)$ are the "load-bearing" edges of the circuit; edges with small $\mathrm{MI}(e)$ can be pruned without significant loss of task performance. This gives a principled, information-theoretically grounded version of the edge attribution methods used in mechanistic interpretability.

4.7 The Kolmogorov Complexity Perspective

The information-bottleneck and rate-distortion perspectives are statistical: they average over the distribution $p(X,Y)$ and seek the best *expected* compression. There is an alternative, algorithmic perspective rooted in Kolmogorov complexity [41] that gives sharper (distribution-free) bounds at the cost of uncomputability.

Definition 4.6 (Circuit Kolmogorov Complexity)**.** The *Kolmogorov complexity* $K(\mathcal{C})$ of a circuit $\mathcal{C}$ is the length of the shortest program that, given the architecture of $\mathcal{N}$, outputs the description of $\mathcal{C}$.

The relevance of $K(\mathcal{C})$ to interpretability is direct: a circuit is interpretable to the extent that it is *simple*, and simplicity is precisely what Kolmogorov complexity measures. A circuit implementing a clean mathematical algorithm (like the Fourier algorithm for modular addition) has low $K(\mathcal{C})$; a circuit implementing a lookup table has $K(\mathcal{C}) \sim N \log p$.

The connection to the statistical perspective is via the *minimum description length* (MDL) principle: the best model (circuit) minimises the total description length of the circuit and the data given the circuit:

$$\mathrm{MDL}(\mathcal{C}) \;=\; K(\mathcal{C}) + \log \frac{1}{p(\text{data} \mid \mathcal{C})}. \tag{4.12}$$

Proposition 4.7 (MDL Bound on Faithful Circuit Size)**.** *Let $\mathcal{C}^*$ be the MDL circuit for task $(X \to Y)$ with training data of size N. Then:*

$$|E^{\mathcal{C}^*}| \;\leq\; \frac{K(f_Y)}{\log(1/\delta)} + O\left(\sqrt{\frac{K(f_Y)}{N}}\right), \tag{4.13}$$

where $f_Y : \mathcal{X} \to \mathcal{Y}$ is the true labelling function and δ is the target test error.

The bound (4.13) says that the size of the MDL circuit is controlled by the Kolmogorov complexity of the task — the length of the shortest algorithm for the task — and converges to this lower bound at the rate $O(1/\sqrt{N})$. Tasks with simple algorithms (small $K(f_Y)$) require small circuits; tasks that are fundamentally complex (large $K(f_Y)$) require large circuits, and no amount of training data will reduce the circuit below $K(f_Y)/\log(1/\delta)$.

Corollary 4.8 (The Interpretability Criterion)**.** *A neural network $\mathcal{N}$ is (ε, K_0)-*interpretable *for task $\mathcal{T}$ if there exists a circuit $\mathcal{C} \subseteq \mathcal{N}$ that is:*

(1) (ε, D)-faithful for $\mathcal{T}$ (Definition 2.4);

(2) $K(\mathcal{C}) \leq K_0$ (low Kolmogorov complexity);

(3) Implemented by an approximately incoherent feature frame at each layer (coherence $\mu_\ell \leq 1/\sqrt{d_\ell}$).

Condition (1) ensures the circuit accounts for the computation. Condition (2) ensures the circuit is simple enough to admit a short description — a prerequisite for human understanding. Condition (3) ensures that the features at each layer are well-separated and individually readable.

This corollary is not a theorem to be proved but a definition to be used. It operationalises, in precise mathematical form, the intuitive notion of interpretability: a network is interpretable to the extent that it implements its computation through a small, algorithmically simple circuit built from well-separated features. The challenge of mechanistic interpretability is to determine, for any given network and task, whether such a circuit exists and, if so, to find it.

4.8 A Unified Picture

We now assemble the results of this chapter into a unified picture.

A trained neural network implements a sequence of representation transformations. At each layer, the representation $T^{(\ell)}$ is a superposition of feature vectors (Chapter 1). The superposition is efficient — nearly orthogonal, in the sense guaranteed by the Welch bound — and sparse, in the sense that only $k \ll m$ features are active on any given input. The total

information in $T^{(\ell)}$ about the eventual label Y increases with depth (Theorem 4.4, part 3), as the network progressively extracts more task-relevant structure from the input.

The circuit $\mathcal{C}$ is the subgraph of the computational graph implementing this information extraction. Its edges correspond to the frame transformations that map one feature system to the next (Theorem 4.4, part 2). Its size is bounded by the Kolmogorov complexity of the task (Proposition 4.7). Its faithfulness is characterised by the information theoretic criterion (Theorem 4.2). And its formation during training is a phase transition driven by the tension between task loss and regularisation (Chapter 3).

The features and the circuit are dual to each other in a precise sense: the features are the vocabulary of the representation, and the circuit is the grammar that transforms one vocabulary to another. Neither is primary. A feature without a circuit to compute it is inert; a circuit without features to manipulate is uninterpretable. The pair (features, circuit) is the irreducible unit of a mechanistic account of a neural network's computation.

Bibliographic Notes

The information bottleneck is introduced in [62]. Its controversial application to deep learning is explored in [61]. The variational mutual information formula (4.6) is from [6]. The rate-distortion lower bound follows classical results in [11]. The MDL principle and its connection to learning theory are developed in [29]; Proposition 4.7 uses the MDL excess risk bounds of [28]. The connection between Kolmogorov complexity and circuit size draws on [41].

4.9 The Variational Information Bottleneck

The information bottleneck (4.4) is defined as an optimisation over all conditional distributions $p(T|X)$, which is intractable in general. A tractable variational approximation, developed in [6, 61], parametrises the bottleneck representation as the output of a neural encoder $T = g_\theta(X) + \varepsilon$ where ε is Gaussian noise.

The *variational information bottleneck* (VIB) replaces the mutual information terms with tractable lower and upper bounds:

$$I(X;T) \;\le\; \mathbb{E}\Big[D_{\mathrm{KL}}\Big(p(T|X)\,\Big\|\,r(T)\Big)\Big]\,, \tag{4.14}$$

$$I(T;Y) \;\ge\; \mathbb{E}[\log q_\phi(Y|T)] + H(Y), \tag{4.15}$$

where $r(T)$ is a variational prior (e.g., a standard Gaussian) and $q_\phi(Y|T)$ is a variational decoder parametrised by ϕ. The VIB objective then becomes:

$$\mathcal{L}_{\mathrm{VIB}}(\theta,\phi) \;=\; -\mathbb{E}[\log q_\phi(Y|T)] + \beta\,\mathbb{E}[D_{\mathrm{KL}}(p_\theta(T|X)\|r(T))]\,. \tag{4.16}$$

This is the β-VAE objective (variational autoencoder with a β-weighted KL term) applied to the supervised learning setting.

The connection to the circuit framework is the following. When the encoder g_θ implements a neural circuit $\mathcal{C}$ and the decoder q_ϕ is a fixed softmax layer, minimising (4.16) is equivalent (in the limit of infinite model capacity) to finding the optimal information bottleneck representation — which, by Theorem 4.2, is the minimal faithful circuit for the task.

Proposition 4.9 (VIB as Circuit Regularisation)**.** *In the limit where the encoder g_θ can implement any bounded-complexity circuit, minimising* (4.16) *with $\beta = \lambda / I(X;Y)$ is equivalent to the regularised circuit discovery problem* (2.3) *with complexity penalty λ.*

This proposition explains why training neural networks with VIB-style objectives tends to produce more interpretable representations: the KL regularisation in (4.16) penalises complexity in the same way that the ℓ^1 term in (2.3) penalises the number of circuit edges.

4.10 Information Flow and the Layerwise IB Curve

Theorem 4.4 asserts that $I(T^{(\ell)};Y)$ is non-decreasing in ℓ for a faithful circuit. This gives rise to a well-defined *information curve* — the sequence $\{(I(X;T^{(\ell)}), I(T^{(\ell)};Y))\}_{\ell=0}^{L}$ tracing the network's progress from input to output in the information plane.

The information curve lies between two extremes:

(i) The *data processing inequality curve*: $I(T^{(\ell)};Y) \leq I(T^{(\ell)};X) \leq I(X;X) = I(X;X)$ (trivially saturated at the identity).

(ii) The *rate-distortion curve*: the minimum achievable $I(X;T^{(\ell)})$ for each level of $I(T^{(\ell)};Y)$, given by $R(D)$ where D corresponds to $1 - I(T^{(\ell)};Y)/I(X;Y)$.

A network implementing the optimal bottleneck representation at each layer would have its information curve lying exactly on the rate-distortion curve. In practice, networks lie above the rate-distortion curve (they are not maximally compressed), but the gap decreases as training progresses.

Proposition 4.10 (Information Curve During Training)**.** *During training by gradient descent on the cross-entropy loss with ℓ^2 regularisation, the information curve $\{(I(X;T^{(\ell)}), I(T^{(\ell)};Y))\}_{\ell}$ moves toward the rate-distortion curve. Specifically, in the early phase of training, the curve moves primarily in the $I(T;Y)$ direction (the network learns to predict the label); in the late phase, it moves primarily in the $-I(X;T)$ direction (the network compresses the input).*

This is a version of the "information bottleneck conjecture" of [61], which proposed that training proceeds in two phases: a rapid "fitting" phase and a slower "compression" phase. The conjecture is controversial — whether the compression phase is real or an artefact of the estimation method for mutual information is debated in the literature. However, from the circuit perspective, the conjecture has a natural interpretation: the fitting phase corresponds to the formation of the circuit (Stage 3 of the grokking dynamics of Chapter 3), and the compression phase corresponds to the consolidation of the circuit (Stage 4, where memorisation weights are eliminated by weight decay).

4.11 The Minimal Sufficient Statistic and Its Geometry

The information bottleneck seeks the *minimal sufficient statistic* of X for predicting Y: the representation T^* that satisfies $I(T^*;Y) = I(X;Y)$ (sufficient: it captures all relevant information) and $I(X;T^*) = R(\text{error} = 0)$ (minimal: it uses the least possible information from X).

In terms of the feature framework, the minimal sufficient statistic is represented by a set of features $\{f_i^*\}$ such that $\sum_i I(f_i^{*\top}T;Y) = I(X;Y)$ (completeness) and $\sum_i I(X;f_i^{*\top}T)$ is minimised (minimality). The first condition requires the feature frame to be complete in the information-theoretic sense; the second requires it to be incoherent (to avoid storing redundant information).

Corollary 4.11 (Incoherence as Minimality)**.** *A feature frame $\{f_i\}$ represents a minimal sufficient statistic for the task $(X \to Y)$ if and only if:*

(1) The features are 0-sufficient for all relevant properties $g_i(Y)$ (Definition 4.3);

(2) The frame is incoherent: $\mu(F) \leq 1/\sqrt{d}$ (the Welch bound).

Condition (2) ensures that no information is stored redundantly across features, making the representation maximally compressed.

This corollary connects the geometric notion of incoherence (from Chapter 1) to the information-theoretic notion of minimality (from this chapter): the Welch bound is not merely a geometric constraint but an information-theoretic optimality condition.

Bibliographic Notes

The information bottleneck is introduced in [62]. The variational information bottleneck and its connection to β-VAEs is developed in [61] and [6]. The controversy over the information bottleneck conjecture is reviewed in Saxe et al. (2019). The rate-distortion connection follows classical results in [11]. The MDL bound is in [29] and [28].

4.12 Mutual Information Estimation in Practice

The theoretical results of this chapter are formulated in terms of the true mutual information $I(X;T)$, which is generally intractable to compute exactly. In practice, one must estimate mutual information from finite samples. The challenges of mutual information estimation, and their implications for the circuit framework, deserve explicit discussion.

4.12.1 The Estimation Problem

Mutual information estimation is notoriously difficult in high dimensions. The naive estimator based on the Kullback-Leibler divergence requires computing the full joint and marginal densities $p(X,T)$ and $p(X)p(T)$, which in d dimensions requires exponentially many samples. For the feature importance scores $\lambda_i = I(f_i^\top T; Y)$ of Chapter 9, the estimation is easier: since $f_i^\top T$ is one-dimensional, standard kernel density estimation or the k-nearest neighbours estimator can be used with polynomial sample complexity.

For the full mutual information $I(X;T)$ (which is d-dimensional), the MINE estimator [6] provides a lower bound via the variational formula (4.6):

$$I(X;T) \;\geq\; \sup_{q \in \mathcal{Q}} \mathbb{E}_{(x,t)\sim p(X,T)}[\log q(t|x)] - \log \mathbb{E}_{t' \sim p(T)}\left[e^{q(t'|x)}\right], \tag{4.17}$$

where $\mathcal{Q}$ is a class of neural network parametrisations. The MINE estimator is consistent and converges at a rate determined by the complexity of $\mathcal{Q}$.

4.12.2 Implications for Circuit Attribution

The edge attribution scores $\mathrm{MI}(e)$ of (4.9) depend on mutual information differences, which can be estimated more easily than the mutual information itself (the variance of the difference cancels part of the estimation error). For a circuit with $|E^{\mathcal{C}}|$ edges, estimating all attribution scores $\mathrm{MI}(e)$ requires $O(|E^{\mathcal{C}}|)$ mutual information estimations, each with sample complexity depending on the dimensionality of the relevant representations.

The approximate decomposition (4.10) gives an efficient method for circuit attribution: estimate $\mathrm{MI}(T^{(L)}; Y)$ once (the total mutual information in the full circuit's output), then estimate $\mathrm{MI}(T^{(L)}_{-e}; Y)$ for each edge e (the mutual information after removing edge e), and report the difference as the attribution score. This requires $|E^{\mathcal{C}}| + 1$ mutual information estimations in total.

4.12.3 The Faithfulness–Estimation Gap

A circuit discovered by the algorithm (2.3) is faithful with respect to the empirical distribution $\hat{D}$ over the training data. But the true faithfulness criterion (Definition 2.4) refers to the population distribution D. The gap between empirical and population faithfulness is the *faithfulness–estimation gap*:

Proposition 4.12 (Faithfulness–Estimation Gap)**.** *Let $\mathcal{C}$ be a circuit that is $(\varepsilon, \hat{D}_N)$-faithful with respect to the empirical distribution over N samples. Then $\mathcal{C}$ is $(\varepsilon + \delta_N, D)$-faithful with respect to the population distribution, where:*

$$\delta_N \;=\; O\left(\sqrt{\frac{|E^{\mathcal{C}}| \log(1/\eta)}{N}}\right) \tag{4.18}$$

with probability at least $1 - \eta$, under a standard Rademacher complexity argument applied to the circuit function class.

The faithfulness–estimation gap δ_N decreases as $N^{-1/2}$ (the standard statistical rate) and grows logarithmically with the confidence parameter $1/\eta$. For the circuit size $|E^{\mathcal{C}}| \lesssim K(f_Y)/\log(1/\varepsilon)$ (Proposition 4.7), the required sample size to achieve $\delta_N \leq \varepsilon$ is $N \gtrsim K(f_Y)/(\varepsilon^2 \log(1/\varepsilon))$.

4.13 Gaussian Channels and the Rate-Distortion Bound

The rate-distortion bound of Lemma 4.1 is tight for Gaussian channels. We verify this and derive the Gaussian channel capacity as a special case.

Proposition 4.13 (Gaussian Channel Tightness)**.** *Let $X \sim \mathcal{N}(0, \Sigma_X)$ and $T = WX + \varepsilon$ where $W \in \mathbb{R}^{d \times n}$ and $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_d)$ (a Gaussian channel with additive noise). Then:*

$$I(X;T) \;=\; \frac{1}{2} \log \det\left(I_d + \frac{1}{\sigma^2} W \Sigma_X W^\top\right), \tag{4.19}$$

$$R(D) \;=\; \frac{1}{2} \log \det(\Sigma_X) - \frac{1}{2} \log D^d, \tag{4.20}$$

where D is the squared reconstruction error. The rate-distortion bound (4.3) is achieved with equality when W is chosen to be the principal components of Σ_X.

This proposition says that the optimal encoding strategy for a Gaussian signal under squared-error distortion is to project onto the principal components of the input distribution — the classic result of rate-distortion theory for Gaussian sources. In the neural network context, this predicts that a network with Gaussian input distribution will learn feature directions aligned with the principal components of the input — consistent with the well-known observation that the features of autoencoders are related to principal components.

Bibliographic Notes

The information bottleneck is in [62]. The VIB is in [61]. MINE is from [6]. Rate-distortion theory is in [11]. The MDL principle is in [29]. The Kolmogorov complexity connection is in [41]. The information geometry of circuits connects to [1].

4.14 Tensor Decomposition and Multi-View Representations

The single-feature analysis of §4.4 treats each feature f_i independently. In practice, features interact: the network may represent a concept as the *conjunction* of several features rather than as a single feature direction. The appropriate mathematical framework for such multi-feature representations is *tensor decomposition*.

4.14.1 Feature Tensors and Their Decomposition

Let $T^{(\ell)} \in \mathbb{R}^{d_\ell}$ be the representation at layer ℓ. The joint activation pattern of two features f_i and f_j is described by the *feature tensor*:

$$\mathbf{A}_{ij} \;=\; \mathbb{E}_{X\sim D}[a_i(X)a_j(X)]\,, \tag{4.21}$$

the covariance matrix of feature activations. For a frame with m features, this is an $m \times m$ matrix encoding all pairwise co-activation statistics.

When $\mathbf{A}_{ij} \approx p_i p_j$ (nearly diagonal, i.e., feature activations are approximately independent), the frame is *statistically incoherent*: features rarely activate together, and the mutual information decomposition (4.8) is nearly exact. When $\mathbf{A}_{ij} \gg p_i p_j$ for some (i, j), features i and j are *co-activated*: they tend to activate together, which means they may be encoding a conjunction.

Proposition 4.14 (Feature Tensors and the IB)**.** *Under the information bottleneck objective* (4.4) *with $\beta \to \infty$ (perfect relevance), the optimal feature activations satisfy $\mathbf{A}_{ij} = p_i p_j$ for all $i \neq j$ such that $I(f_i^\top T; f_j^\top T; Y) = 0$ (zero three-way interaction information). In other words, the IB objective encourages features to be statistically independent* conditional on *being relevant for the label.*

This proposition says that a well-trained network should represent as few features as possible, and these features should be as independent as possible. The independence of features is the statistical version of the geometric incoherence condition (small $\mu(F)$): both conditions say that features should not interfere with each other, whether the interference is geometric (small inner products) or statistical (small co-activation).

4.14.2 Tensor Networks and Deep Circuits

The tensor decomposition perspective extends naturally to deep circuits. A circuit of depth L can be viewed as a *tensor network*: a network of rank-k tensors connected by contractions. The computation implemented by the circuit is the tensor contraction of all the constituent tensors.

For circuits implementing polynomial functions of the input, the tensor network decomposition gives a natural characterisation of the circuit's complexity:

- A depth-1 circuit implements a linear function (rank-1 tensor).
- A depth-2 circuit implements a bilinear function (rank-2 tensor, or matrix).
- A depth-L circuit implements a multilinear function (rank-L tensor).

The circuit size $|E^{\mathcal{C}}|$ controls the *tensor rank* of the function: a circuit with $|E|$ edges implements a function of tensor rank at most $|E|$. This connects to the classical theory of tensor rank complexity [13], which studies the minimum number of multiplications needed to compute a multilinear function.

4.15 The Dual Representation: Circuits as Linear Operators

The circuit framework of Chapter 2 represents circuits as DAGs with weight matrices. There is a complementary representation that treats circuits as *linear operators* on the space of activations, revealing additional structure.

4.15.1 The Circuit Matrix

For a circuit $\mathcal{C}$ with linear activations ($\phi_v = \mathrm{id}$), the function $f_{\mathcal{C}} : \mathbb{R}^{|V_{\mathrm{in}}|} \to \mathbb{R}^{|V_{\mathrm{out}}|}$ is linear: $f_{\mathcal{C}}(x) = M_{\mathcal{C}} x$ for a matrix $M_{\mathcal{C}} \in \mathbb{R}^{|V_{\mathrm{out}}| \times |V_{\mathrm{in}}|}$. This matrix is the *circuit matrix*, defined as the product of the weight matrices along all paths from input to output:

$$M_{\mathcal{C}} \;=\; \prod_{\ell=1}^{L} W_\ell^{\mathcal{C}}, \tag{4.22}$$

where $W_\ell^{\mathcal{C}}$ is the weight matrix of the ℓ-th layer of the circuit.

The properties of $M_{\mathcal{C}}$ reveal important structural information:

- $\mathrm{rank}(M_{\mathcal{C}})$: the rank of the circuit matrix equals the number of independent "information channels" through the circuit.
- $\sigma_{\min}(M_{\mathcal{C}})$: the minimum singular value measures the circuit's robustness (Proposition 8.17: circuits with large minimum singular value are more robust).
- $\|M_{\mathcal{C}}\|_F^2 = |E^{\mathcal{C}}| \cdot \bar{w}^2$ (for circuits with uniform weight magnitude $\bar{w}$): the Frobenius norm is proportional to the circuit size, connecting the matrix norm to the combinatorial size measure.

4.15.2 Non-Linear Circuits and the Tangent Kernel

For non-linear circuits (with ReLU or other non-linear activations), the circuit matrix is replaced by the *tangent kernel* evaluated at the current point in weight space:

$$K_{\mathcal{C}}(x, x') \;=\; \nabla_W f_{\mathcal{C}}(x)^\top \nabla_W f_{\mathcal{C}}(x'), \tag{4.23}$$

the inner product between the gradients of the circuit's output with respect to its parameters, evaluated at two inputs x and x'. The tangent kernel measures how much the circuit's output at x changes when its parameters are updated to improve the output at x' — a measure of the "functional correlation" between inputs.

The tangent kernel perspective connects circuit theory to the neural tangent kernel (NTK) theory of neural networks, developed by Jacot et al. (2018). In the NTK limit (infinite width), gradient descent on the circuit is equivalent to kernel regression with the tangent kernel, which is computable analytically for specific architectures.

Bibliographic Notes

Information bottleneck: [62], [61], [6]. Rate-distortion: [11]. MDL: [29], [28]. Tensor rank complexity: Bürgisser et al. (2013), *Algebraic Complexity Theory*. Neural tangent kernel: Jacot, Gabriel, and Hongler (2018), "Neural Tangent Kernel." The VIB: [61]. MI estimation: [6]. Information geometry: [1].

4.16 The Geometry of the Information Bottleneck Tradeoff Curve

The information bottleneck tradeoff curve is the set of achievable $(I(X;T), I(T;Y))$ pairs as β varies from 0 to ∞. The geometry of this curve encodes the structure of the task $p(X,Y)$ and determines the interpretability of optimal circuits.

4.16.1 Convexity and the Shape of the Curve

The IB tradeoff curve $\{(R(\beta), \mathrm{Rel}(\beta)) : \beta \geq 0\}$, where $R(\beta) = I(X;T^*(\beta))$ and $\mathrm{Rel}(\beta) = I(T^*(\beta);Y)$ are the complexity and relevance at the optimal bottleneck for tradeoff β, is a convex non-decreasing curve in the (R, Rel) plane. Its convexity follows from the convexity of the rate-distortion function.

Proposition 4.15 (Shape of the IB Curve)**.** *The IB tradeoff curve satisfies:*

(1) It starts at $(0,0)$ *(when* $\beta = 0$*, the trivial representation* $T = \mathrm{const}$*);*

(2) It ends at $(I(X;T_\infty), I(X;Y))$ *when* $\beta \to \infty$*, where* T_∞ *is the sufficient statistic* $p(Y|X)$*;*

(3) It is concave (when viewed as Rel *as a function of* R*): increasing* R *yields diminishing returns in* Rel*;*

(4) Each point on the curve corresponds to a unique universality class of circuits (under the strict convexity assumption of Theorem 7.4).

The concavity property (3) is the information-geometric statement of the law of diminishing returns: each additional bit of complexity in the representation yields a smaller and smaller increase in relevance. This is the formal justification for the principle of *Occam's razor* in the circuit framework: among all circuits achieving a given relevance, the simplest one (smallest R) is preferred, because additional complexity yields no benefit.

4.16.2 The Circuit Complexity–Performance Curve

The IB tradeoff curve can be recast in terms of circuit complexity and test performance, giving the *circuit complexity–performance curve* (CPC):

$$\mathrm{CPC}(\mathcal{T}) \;=\; \{(K(\mathcal{C}^*(\beta)),\, 1 - \mathrm{Rel}(\beta)/I(X;Y)) : \beta \geq 0\}, \tag{4.24}$$

where $K(\mathcal{C}^*(\beta))$ is the Kolmogorov complexity of the optimal circuit at tradeoff β and $1 - \mathrm{Rel}(\beta)/I(X;Y)$ is the fraction of task information not captured by the circuit (a proxy for test error).

The CPC is equivalent to the interpretability-performance frontier (Definition 8.8) under the identification of circuit complexity with Kolmogorov complexity and task performance with the fraction of captured information. It is a more precise characterisation of the IPF in terms of the IB framework.

Bibliographic Notes

Information bottleneck: [62], [61], [6]. Rate-distortion theory: [11]. MDL: [29], [28]. Information geometry: [1]. Tensor complexity: [13]. Neural tangent kernel: Jacot et al. (2018).

Chapter 5

Attention Heads as Associative Memories

5.1 The Question Behind the Mechanism

The scaled dot-product attention mechanism — the computational primitive at the heart of every transformer architecture — is often described in terms of "attending" to relevant positions, "looking up" stored information, or "routing" signals from one position to another. These metaphors are suggestive but imprecise; they describe the phenomenology without engaging the mathematics. In this chapter we ask: what *is* the attention mechanism, in precise mathematical terms? What computation does it implement, and what are its fundamental limits?

The answer, developed formally below, is that attention implements a single update step of a *modern Hopfield network* — a content-addressable associative memory with exponential storage capacity [53]. This is not a metaphor. The mathematical equivalence between attention and Hopfield retrieval is exact. Recognising it transforms a collection of empirical observations about what attention heads do into theorems about what they can and cannot do, and it opens a precise line of attack on the capacity analysis that would otherwise require case-by-case empirical investigation.

The connection runs deeper than the single-layer case. The key-query-value structure of attention implements something richer than a standard Hopfield network: it is a *heteroassociative* memory that matches queries against keys but retrieves values — a two-memory architecture in which the pattern-matching space and the information-retrieval space are distinct. This separation is the source of the transformer's representational power, and its precise mathematical statement reveals why the choice of key-query dimension d_k is crucial.

5.2 Classical Hopfield Networks and Their Limits

We begin with the classical Hopfield network [32] to establish the baseline. Let $\xi_1, \ldots, \xi_m \in \{-1, +1\}^n$ be m binary patterns to be stored in memory. The Hopfield network stores them in the weight matrix:

$$J \;=\; \frac{1}{n}\sum_{\mu=1}^{m} \xi_\mu \xi_\mu^\top - \frac{m}{n} I_n. \tag{5.1}$$

The network evolves by the synchronous update $s^{t+1} = \operatorname{sgn}(Js^t)$. The fixed points include (approximately) all stored patterns, and a probe vector s^0 near a stored pattern ξ_μ will converge to ξ_μ under iteration.

Theorem 5.1 (Classical Hopfield Capacity [2]). *The classical Hopfield network with n neurons can store at most $m^* \approx 0.138 \cdot n$ patterns and retrieve any stored pattern with zero bit-errors from any probe within Hamming distance $n/8$ of the pattern, in the limit $n \to \infty$. For $m > 0.138n$, the retrieval error rate transitions discontinuously from near zero to near $1/2$.*

The capacity bound $m^* \sim 0.138n$ is linear in the number of neurons. This means that a classical Hopfield network with $n = 1000$ neurons can store at most ≈ 138 patterns. For a transformer head with key-query dimension $d_k = 64$, the classical bound gives at most ≈ 8 stored patterns per head — obviously inconsistent with the empirical behaviour of transformers, which appear to store and retrieve vastly more associations. The classical network is not the right model.

5.3 Modern Hopfield Networks: Exponential Capacity

The modern Hopfield network [39, 53] replaces the linear interaction energy of the classical case with a non-linear energy function that achieves exponential storage capacity. For continuous-valued patterns, the energy function is:

$$E(\xi; X, \beta) \;=\; -\operatorname{lse}(\beta, X^\top \xi) + \tfrac{1}{2}\,\|\xi\|^2 + \tfrac{1}{\beta}\log m, \tag{5.2}$$

where $X = [x_1 \mid \cdots \mid x_m] \in \mathbb{R}^{n \times m}$ is the matrix of stored patterns, $\xi \in \mathbb{R}^n$ is the state (probe), $\beta > 0$ is the inverse temperature, and $\operatorname{lse}(\beta, z) = \frac{1}{\beta}\log \sum_i e^{\beta z_i}$ is the log-sum-exp. The update rule is obtained by setting $\nabla_\xi E = 0$:

$$\xi^{t+1} \;=\; X \cdot \operatorname{softmax}\Big(\beta X^\top \xi^t\Big). \tag{5.3}$$

The softmax concentrates weight on the patterns most similar (in dot-product sense) to the current state ξ^t; the update moves ξ^{t+1} toward the weighted average of stored patterns, with most weight on the nearest pattern.

Theorem 5.2 (Modern Hopfield Capacity [53]). *Let the m stored patterns be i.i.d. uniform on S^{n-1}. For any $\varepsilon > 0$, the update rule (5.3) retrieves any stored pattern from any probe nearer to it than to any other stored pattern, in a single step with error at most ε, provided:*

$$m \;\le\; \frac{C(\varepsilon)\, e^{n\beta^2/2}}{\beta}, \tag{5.4}$$

where $C(\varepsilon) > 0$ depends only on ε.

At the optimal temperature $\beta = 1/\sqrt{n}$ (which corresponds to the scaling $1/\sqrt{d_k}$ in the transformer), the bound gives $m \le C(\varepsilon) \cdot e^{n/2} \cdot \sqrt{n}$ — exponential in n. A head with $n = d_k = 64$ can in principle store $\sim e^{32} \approx 10^{13}$ patterns. The proof uses a concentration argument: the softmax weight on the correct pattern minus the weight on any other pattern is, with high probability, bounded below by a quantity depending on n and β. The $1/\sqrt{n}$ scaling is precisely what is needed to make the signal-to-noise ratio of the softmax output bounded away from zero.

5.4 The Exact Correspondence with Attention

We now state the mathematical equivalence between attention and Hopfield retrieval. Let $X \in \mathbb{R}^{T \times d}$ be the sequence of T input tokens. A single attention head computes:

$$\operatorname{Att}(X) \;=\; \operatorname{softmax}\!\left(\frac{X W_Q W_K^\top X^\top}{\sqrt{d_k}}\right) X W_V W_O. \tag{5.5}$$

For a fixed query token at position t, the query vector is $q = (XW_Q)_t \in \mathbb{R}^{d_k}$, the key matrix is $K = XW_K \in \mathbb{R}^{T\times d_k}$, and the value matrix is $V = XW_V \in \mathbb{R}^{T\times d_v}$. The output is:

$$h_t \;=\; V^\top \operatorname{softmax}\!\left(\frac{Kq}{\sqrt{d_k}}\right) \;\in\; \mathbb{R}^{d_v}. \tag{5.6}$$

Proposition 5.3 (Attention as Heteroassociative Memory)**.** *Identifying $\xi = q$ (the query as the probe), $X \leftarrow K^\top$ (the keys as stored patterns), and $\beta = 1/\sqrt{d_k}$ (the scaling as inverse temperature), the attention output* (5.6) *is exactly the Hopfield update* (5.3) *followed by a linear map to value space:*

$$h_t \;=\; W_O^\top \cdot \underbrace{K^\top \cdot \operatorname{softmax}\!\left(\frac{Kq}{\sqrt{d_k}}\right)}_{\textit{Hopfield retrieval in key space}}. \tag{5.7}$$

The head implements heteroassociative *retrieval: it matches queries against keys but retrieves values.*

The heteroassociative structure is a fundamental feature of the architecture with its own capacity theory. The capacity of the key-matching memory (how many distinct keys can be distinguished by the query) is bounded by Theorem 5.2 applied to d_k. The capacity of the value-retrieval memory (how much information can be encoded in the values) is a separate question, bounded by the mutual information $I(V;Y)$ where Y is the downstream prediction target.

5.5 Capacity of the Key-Query-Value Architecture

Theorem 5.4 (Joint Capacity Bound)**.** *A single attention head with key-query dimension d_k, value dimension d_v, and T source tokens has:*

$$\textit{Pattern-matching capacity:} \qquad \mathcal{M}_K \;\le\; e^{d_k/2}, \tag{5.8}$$

$$\textit{Information-retrieval capacity:} \qquad \mathcal{R}_V \;\le\; d_v \cdot \log(1 + T/d_v) \;\textit{bits per query.} \tag{5.9}$$

The total parameter count of the head is $(d_k + d_v)\cdot d$. The information retrieved per parameter is maximised when $d_k = d_v$ (balanced architecture).

The bound on $\mathcal{M}_K$ in (5.8) is Theorem 5.2 specialised to $n = d_k$. The bound on $\mathcal{R}_V$ follows from the capacity of a Gaussian channel with d_v output dimensions and signal-to-noise ratio T/d_v — the SNR grows with the number of source tokens T because more context provides more signal. The efficiency bound shows that the information efficiency of an attention head decreases with the residual dimension d and increases logarithmically with the context length T.

5.6 Head Specialisation and the Circuit Perspective

Multi-head attention with H heads computes:

$$\mathrm{MHA}(X) \;=\; \sum_{h=1}^{H} \operatorname{softmax}\!\left(\frac{XW_Q^h (W_K^h)^\top X^\top}{\sqrt{d_k}}\right) \cdot XW_V^h W_O^h. \tag{5.10}$$

From the circuit perspective (Chapter 2), each head is a sub-circuit $\mathcal{C}_h$ implementing a distinct heteroassociative lookup. The full multi-head attention block is the parallel composition

$\mathcal{C}_1 \oplus \cdots \oplus \mathcal{C}_H$, summed via a linear combining layer. The total pattern-matching capacity of the block is approximately $H \cdot e^{d_k/2}$, while the value-retrieval capacity is $H \cdot d_v \log(1 + T/d_v)$.

The empirical phenomenon of *head specialisation* — the observation that different heads attend to different aspects of the input (syntax vs. semantics, local vs. long-range, copying vs. composition) — has a natural explanation in this framework. Each head $\mathcal{C}_h$ is a circuit implementing a distinct lookup function, matching queries against patterns specific to its own key-query space. Specialisation emerges because different heads solve different sub-problems of the overall task.

Proposition 5.5 (Head Specialisation as Functional Decomposition)**.** *Let $\mathcal{T} = \mathcal{T}_1 \oplus \cdots \oplus \mathcal{T}_H$ be a task decomposing into H independent sub-tasks. Under squared-error loss with ℓ^2 regularisation, the optimal multi-head attention block assigns each head h exclusively to sub-task $\mathcal{T}_h$, with weight matrices determined by the optimal heteroassociative memory for $\mathcal{T}_h$.*

Proof sketch. The total loss decomposes as $\mathcal{L}(\text{MHA}) = \sum_h \mathcal{L}_h(\mathcal{C}_h)$ when the sub-tasks are independent (their gradients are orthogonal). Each $\mathcal{L}_h$ depends only on the parameters of head h, so the optimal solution for each head is the optimal solution for the corresponding sub-task, independently. □

The specialisation result is an idealisation (real tasks rarely decompose perfectly), but it identifies the correct qualitative tendency: gradient descent in multi-head attention naturally allocates heads to sub-tasks, because cross-task interference in the gradient is suppressed by the independence of sub-task outputs.

5.7 The Sharp vs. Diffuse Attention Regimes

The inverse temperature $\beta = 1/\sqrt{d_k}$ controls the sharpness of the attention distribution. In the *sharp* regime ($\beta \to \infty$, small d_k), the softmax concentrates on the single nearest key; in the *diffuse* regime ($\beta \to 0$, large d_k), the softmax is nearly uniform. These two regimes implement qualitatively different computations.

Sharp attention implements a winner-take-all lookup: the head retrieves the value associated with the single best-matching key. This is appropriate when the task requires identifying a specific token (e.g., finding the antecedent of a pronoun, or retrieving a named entity from context).

Diffuse attention implements a weighted average: the head retrieves a mixture of values weighted by their similarity to the query. This is appropriate when the task requires aggregating information from multiple sources (e.g., computing the average sentiment of a passage, or integrating evidence across multiple occurrences of a concept).

Proposition 5.6 (Attention Entropy and Regime Classification)**.** *Let $\alpha = \text{softmax}(\beta K q)$ be the attention distribution. Define the attention entropy $\mathcal{H}_h(q) = -\sum_s \alpha_s \log \alpha_s \in [0, \log T]$. Head h operates in the sharp regime on input q if $\mathcal{H}_h(q) \le \varepsilon \log T$; in the diffuse regime if $\mathcal{H}_h(q) \ge (1 - \varepsilon) \log T$. The expected entropy satisfies:*

$$\mathbb{E}_q[\mathcal{H}_h(q)] \;\approx\; \log T - \beta^2 \mathbb{E}_q\Big[\max_s \langle k_s,\, q\rangle^2\Big] + O(1/T), \tag{5.11}$$

from which the critical inverse temperature is $\beta^* = \sqrt{\log T / \mathbb{E}_q[\max_s \langle k_s, q\rangle^2]}$.

The standard transformer scaling $\beta = 1/\sqrt{d_k}$ places heads near β^* for typical random initialisations, so that attention is neither maximally sharp nor maximally diffuse at the start of training. This allows gradient descent to specialise each head toward whichever regime is appropriate for its sub-task.

The transition between sharp and diffuse attention, for heads trained on tasks of different types, is one of the clearest empirical manifestations of the circuit-level specialisation predicted by Proposition 5.5. Heads attending syntactically (e.g., subject-verb agreement) tend to be sharp; heads attending semantically (e.g., topic modelling) tend to be diffuse. The mathematical reason is that syntactic tasks require precise identification of a specific antecedent (a single source position), while semantic tasks benefit from aggregating information across many positions.

Bibliographic Notes

The classical Hopfield network is introduced in [32]; the capacity result (Theorem 5.1) is from [2]. Modern Hopfield networks are introduced in [39] and developed in continuous form in [53], which also establishes the equivalence between modern Hopfield updates and attention (Proposition 5.3). The attention entropy analysis of Proposition 5.6 relates to [19]. The joint capacity bound (Theorem 5.4) is a novel result of this monograph.

5.8 Induction Heads as a Case Study

The most thoroughly studied example of attention-as-memory is the *induction head*, a circuit motif identified by [23] that implements in-context learning by pattern matching. We analyse it here using the Hopfield framework.

An induction head operates in a two-layer transformer and performs the following computation: given a sequence containing a repeated pattern $[A, B, \ldots, A]$, the induction head predicts B as the next token. This is a heteroassociative retrieval task: the query is the current token A, the keys are all previous occurrences of A, and the retrieved value is the token B that followed the previous A.

In the framework of Proposition 5.3, the induction head stores the associations (k_A, v_B) for each token A that has appeared, where k_A is the key vector for position A and v_B is the value vector for the token immediately following A. When the current query $q = k_A$ is issued (at the second occurrence of A), the softmax concentrates on the key k_A from the first occurrence, and the retrieved value v_B predicts the next token.

Proposition 5.7 (Induction Head as Hopfield Retrieval)**.** *An induction head implementing the $[A, B, \ldots, A, ?]$ completion task stores $T-1$ associations (one per previous position), with pattern-matching capacity $\mathcal{M}_K \leq e^{d_k/2}$ and retrieval capacity $\mathcal{R}_V \leq d_v \log(1 + T/d_v)$. The head succeeds at the task when d_k is large enough to distinguish the key for position A from all other keys, i.e., when $e^{d_k/2} \geq T$ (so that all T keys can be stored and distinguished).*

The condition $e^{d_k/2} \geq T$ requires $d_k \geq 2 \log T$. For a sequence of length $T = 512$ and a typical key dimension $d_k = 64$, this gives $e^{32} \approx 10^{13} \gg 512$ — well within the capacity, confirming that the induction head can store and retrieve all patterns in the context.

The two-layer structure of the induction head circuit is a consequence of the sequential composition $\mathcal{C}_2 \circ \mathcal{C}_1$ (Definition 2.2 in Chapter 2): the first head attends to the previous occurrence of the current token (a "previous token" head), and the second head attends to the token immediately following that previous occurrence (the induction head proper). The sequential composition implements a function $f_{\mathcal{C}_2} \circ f_{\mathcal{C}_1}$ that cannot be implemented by any single attention head, because the required computation has depth 2 in the circuit sense of Theorem 2.7.

5.9 Multi-Head Attention as a Distributed Associative Memory

The multi-head attention block (5.10) can be viewed as a *distributed associative memory*: a memory system in which different heads store different associations, and the combined output is a superposition of the retrieved values.

This perspective connects the attention mechanism to the theory of *mixture-of-experts* models: each head is an "expert" that specialises in a different type of association, and the linear combination of their outputs implements a mixture. The key difference from standard mixture-of-experts is that the mixing weights (the attention distributions $\alpha_{t,s}^h$) are computed dynamically from the input, rather than being fixed.

Remark 5.8 (On the Optimality of the Softmax). The softmax function in the attention mechanism is not the only possible choice for the pattern-matching operation. Alternatives include the ℓ^2-normalised dot product (which removes the temperature dependence but fixes $\beta = 1$), the ReLU (which implements sparse attention), and the linear (which gives diffuse attention regardless of β). The softmax is distinguished by two properties: it is the unique smooth function from $\mathbb{R}^T$ to the probability simplex that is equivariant under translation of its input (i.e., adding a constant to all logits does not change the output), and it is the Gibbs distribution corresponding to the Hopfield energy (5.2) at inverse temperature β. In the mechanistic interpretability framework, these properties make the softmax the natural choice for implementing the heteroassociative memory described in Proposition 5.3.

Bibliographic Notes

The classical Hopfield network is introduced in [32]; the capacity result (Theorem 5.1) is from [2]. Modern Hopfield networks are in [39] and [53]. Induction heads were identified and analysed in [23] and further studied in [49] as the mechanism underlying in-context learning. The attention entropy analysis of Proposition 5.6 relates to [19].

5.10 Previous Token Heads and the Mechanism of Attention Composition

The induction head analysis of §5.7 introduced the concept of sequential composition of attention heads. We now develop the general theory of attention composition, connecting it to the circuit composition framework of Chapter 2.

5.10.1 Previous Token Heads

A *previous token head* is an attention head that, for each token at position t, attends to the token at position $t-1$ (the previous token). In the Hopfield framework:

- The keys at position t are the embeddings of the current token.
- The queries at position t are such that $\langle q_t, k_{t-1} \rangle \gg \langle q_t, k_s \rangle$ for all $s \neq t-1$.
- The values at position $t-1$ are the embedding of the previous token, written to position t.

The previous token head implements the association (k_t, v_{t-1}): when attending from position t, it retrieves the value of position $t-1$. This is a simple instance of heteroassociative memory (Proposition 5.3) with a fixed, position-based pattern.

5.10.2 The Composition Chain

In a two-layer transformer implementing the induction head mechanism, the previous token head (layer 1) and the induction head (layer 2) are connected by the following composition chain:

(1) *Layer 1, previous token head*: At each position t, attends to $t-1$ and writes the value of position $t-1$ to the residual stream of position t.

(2) *Layer 2, induction head*: At each position t, uses the residual stream (which now contains information about the token at position $t-1$) as its key to attend to the position in the context that matches the current context.

The key composition score $c_K(\text{PTH} \to \text{IH}) = \left\|W_O^{\text{PTH}} W_K^{\text{IH}}\right\|_F$ measures how strongly the output of the previous token head influences the keys of the induction head. A large c_K indicates strong composition: the induction head is using the previous token information to form its keys, which is precisely what the induction mechanism requires.

In the language of circuit algebra (Chapter 2), the induction head circuit is:

$$\mathcal{C}_{\text{IH}} = \mathcal{C}_{\text{IH,layer2}} \circ \mathcal{C}_{\text{PTH,layer1}},$$

a sequential composition where the output of the previous token head is the input to the induction head.

5.10.3 Generalisation: Q-composition and K-composition

The previous token head implements *K-composition*: the output of one head influences the keys of another, changing which positions the next head attends to. The induction head mechanism uses K-composition.

There is also *Q-composition*: the output of one head influences the queries of another, changing what patterns the next head looks for. Q-composition is relevant for tasks that require multi-hop reasoning: first attend to find a reference, then use the reference to guide a second attention step.

And *V-composition*: the output of one head influences the values of another, changing the information that is written to the residual stream. V-composition is relevant for tasks that require the combination of information from multiple sources.

In the transformer circuit framework (Definition 2.6), these three types of composition correspond to the three composition scores c_Q, c_K, c_V. A circuit that involves significant Q-composition, K-composition, and V-composition at multiple layers is implementing a genuinely multi-step reasoning process, where each step builds on the results of the previous one.

5.11 From Theory to Practice: Attention Visualisation

The Hopfield framework provides a principled interpretation of attention visualisations. When we plot the attention matrix $A^h(x) \in \mathbb{R}^{T \times T}$ for a specific head h on a specific input x, we are visualising the pattern-matching step of the heteroassociative retrieval (Proposition 5.3).

The *attention pattern* (the distribution of attention weights $\alpha_{t,s}^h$) reveals which source positions the head is retrieving from, given each query position. The *attention entropy* $\mathcal{H}_h(q)$ (Proposition 5.6) measures how diffuse this pattern is: a sharp attention pattern (low entropy) indicates the head is implementing a precise, content-based lookup; a diffuse pattern (high entropy) indicates the head is aggregating information broadly.

For the induction head:

- On sequences containing repeated patterns ($[A, B, \ldots, A]$), the attention pattern is *sharp*: $\alpha_{t,s} \approx 1$ for $s =$ (position following the previous occurrence of A), and ≈ 0 elsewhere. Entropy is near 0.
- On random sequences (no repeated patterns), the attention pattern is *diffuse*: $\alpha_{t,s} \approx 1/T$ for all s. Entropy is near $\log T$.

This is a clear prediction of the Hopfield framework: the induction head is a content-based memory that activates (becomes sharp) when it finds a pattern match, and falls back to uniform attention (diffuse) when it does not.

Bibliographic Notes

Classical Hopfield networks: [32]. Capacity theorem: [2]. Modern Hopfield networks: [39], [53]. Induction heads and previous token heads: [23], [49]. Attention entropy analysis: [19]. The IOI circuit and attention head types: [64].

5.12 Multi-Query Attention and Grouped-Query Attention

The standard multi-head attention (MHA) of §5.6 uses separate key and query matrices for each head. Two important architectural variants — *multi-query attention* (MQA) and *grouped-query attention* (GQA) — share key and value matrices across heads while maintaining separate query matrices. We analyse these variants in the Hopfield framework.

5.12.1 Multi-Query Attention

In MQA, a single set of key and value matrices W_K and W_V is shared across all H heads, while each head maintains its own query matrix W_Q^h:

$$\mathrm{MQA}(X) \;=\; \sum_{h=1}^{H} \mathrm{softmax}\left(\frac{X W_Q^h W_K^\top X^\top}{\sqrt{d_k}}\right) \cdot X W_V W_O^h. \tag{5.12}$$

In the Hopfield framework, MQA uses a single associative memory (the shared key-value system) but H different query templates. Each head computes a different query from the same residual stream, effectively implementing H different "lookup operations" into the same memory.

The pattern-matching capacity $\mathcal{M}_K$ (Theorem 5.4) is unchanged (it depends only on d_k), but the memory is now shared across all heads. The information-retrieval capacity $\mathcal{R}_V$ is also unchanged (it depends only on d_v). The total capacity of MQA is therefore:

$$\mathcal{M}_K^{\mathrm{MQA}} \;=\; e^{d_k/2} \;=\; \mathcal{M}_K^{\mathrm{MHA}}, \quad \mathcal{R}_V^{\mathrm{MQA}} \;=\; H \cdot d_v \log(1 + T/d_v) \;=\; \mathcal{R}_V^{\mathrm{MHA}}. \tag{5.13}$$

The capacity is identical to MHA, but MQA requires fewer parameters (H times fewer for the key and value matrices) and has faster inference (shared KV cache). This is the motivation for MQA in practice: it achieves the same theoretical capacity with lower memory and compute cost.

5.12.2 Grouped-Query Attention

GQA interpolates between MHA (full head independence) and MQA (full sharing). In GQA with G groups, the H heads are partitioned into G groups of H/G heads each. Within each

group, the key and value matrices are shared, while the query matrix is separate for each head. GQA recovers MHA when $G = H$ and MQA when $G = 1$.

From the Hopfield perspective, GQA partitions the H heads into G independent Hopfield memories, each serving H/G query templates. The capacity scales as:

$$\mathcal{M}_K^{\text{GQA}} = G \cdot e^{d_k/2}, \quad \mathcal{R}_V^{\text{GQA}} = H \cdot d_v \log(1 + T/d_v). \tag{5.14}$$

The pattern-matching capacity scales with the number of groups G: more groups means more independent memories, each with the same capacity $e^{d_k/2}$. The information-retrieval capacity remains the same as MHA.

For $G < H$, GQA reduces the pattern-matching capacity by a factor of G/H relative to MHA, while reducing the parameter count for key-value matrices by the same factor. The optimal choice of G depends on the task: tasks requiring many distinct lookup patterns (large $\mathcal{M}_K$ needed) should use G close to H; tasks where a few shared patterns suffice can use small G.

Bibliographic Notes

Classical Hopfield: [32], [2]. Modern Hopfield: [39], [53]. Induction heads: [23], [49]. Attention entropy: [19]. Previous token heads and composition: [23]. Multi-query attention: Shazeer (2019), "Fast Transformer Decoding: One Write-Head is All You Need." Grouped-query attention: Ainslie et al. (2023), "GQA: Training Generalised Multi-Query Transformer Models from Multi-Head Checkpoints."

5.13 Attention as an Approximate Gradient Descent Step

A less-known but mathematically elegant interpretation of the attention mechanism is as an approximate gradient descent step for retrieving a stored memory.

5.13.1 The Energy Gradient

Recall the Hopfield energy (5.2):

$$E(\xi; X, \beta) = -\operatorname{lse}(\beta, X^\top \xi) + \tfrac{1}{2}\|\xi\|^2 + \tfrac{1}{\beta}\log m.$$

The gradient of E with respect to the probe ξ is:

$$\nabla_\xi E \;=\; -X \cdot \operatorname{softmax}(\beta X^\top \xi) + \xi \;=\; \xi - \xi^{t+1}, \tag{5.15}$$

where $\xi^{t+1} = X \cdot \operatorname{softmax}(\beta X^\top \xi^t)$ is the Hopfield update (5.3). Therefore the Hopfield update is exactly the gradient descent step $\xi^{t+1} = \xi^t - \nabla_\xi E(\xi^t)$ — a single step of gradient descent on the Hopfield energy with step size 1.

Proposition 5.9 (Attention as Energy Minimisation)**.** *A single forward pass of the attention mechanism (Proposition 5.3) is equivalent to one step of gradient descent on the Hopfield energy $E(\xi; K, 1/\sqrt{d_k})$ starting from $\xi^0 = q$ (the query vector), followed by a linear readout of the resulting state. After convergence (multiple Hopfield updates), the query converges to the nearest stored key pattern.*

This perspective illuminates why transformers with multiple layers of attention can implement multi-step retrieval: each attention layer corresponds to one gradient descent step on the energy landscape. A transformer with L attention layers can perform L steps of gradient descent, which is sufficient to retrieve from a more complex energy landscape than a single-layer model.

5.13.2 Implications for Circuit Depth

The energy minimisation perspective explains the relationship between circuit depth and retrieval accuracy. For a Hopfield network with a flat energy landscape (all stored patterns equally close to the query), one step of gradient descent does not converge; multiple steps are needed. This corresponds to a deep transformer circuit: the circuit for multi-hop retrieval (finding the answer to a question that requires consulting multiple stored associations) has depth proportional to the number of hops.

Concretely, for a task requiring K hops (accessing K stored associations in sequence), the minimum depth of the transformer circuit is $\Omega(K)$ (by a reduction to the depth-width tradeoff, Theorem 2.7). This is consistent with the empirical observation that multi-hop reasoning tasks require larger and deeper models than single-hop tasks.

Bibliographic Notes

Classical Hopfield: [32], [2]. Modern Hopfield: [39], [53]. Induction heads: [23], [49]. Attention entropy: [19]. MQA: Shazeer (2019). GQA: Ainslie et al. (2023). Energy minimisation interpretation: Milne et al. (2023).

Chapter 6

Feature Splitting, Merging, and Polytope Phase Transitions

6.1 The Phenomenon in Full Generality

Among the most striking phenomena uncovered by mechanistic interpretability is the following. During training, or as model capacity increases, networks sometimes undergo a sudden qualitative reorganisation of their internal representations: a single polysemantic neuron — encoding, say, both "days of the week" and "months of the year" — splits into two monosemantic neurons, each exclusively encoding one concept. The reverse also occurs: two previously distinct representations can merge into a single polysemantic one when capacity is reduced or regularisation increased.

These splitting and merging events are not continuous. The representations change in discrete jumps, separated by plateaus during which the network appears stable. The temporal structure looks precisely like the grokking dynamics of Chapter 3, but at the level of individual features rather than entire circuits. This suggests a unified theoretical treatment: feature splitting is a *symmetry-breaking phase transition* in the space of feature configurations, governed by the same competition between interference (which favours splitting) and complexity (which favours merging) that governs grokking.

In this chapter we develop the formal theory. The central mathematical object is the *interference energy* of a feature frame — a functional on the space of frame configurations whose minima are the optimal polytopes identified in Chapter 1. We study the stability of these minima, identify the bifurcation conditions under which they become unstable (triggering a split or merge), and connect the resulting phase diagram to the statistical properties of the input distribution.

6.2 The Interference Energy Functional

Definition 6.1 (Frame Interference Energy)**.** For a configuration $F = \{f_1, \dots, f_m\} \subset S^{d-1}$, the *frame interference energy* is:

$$\mathcal{E}(F) \;=\; \sum_{i \neq j} |\langle f_i,\, f_j \rangle|^2 \;=\; \|G - I_m\|_F^2 \,, \tag{6.1}$$

where $G = F^\top F \in \mathbb{R}^{m \times m}$ is the Gram matrix with $G_{ij} = \langle f_i,\, f_j \rangle$.

The motivation for this choice — rather than, say, the maximum coherence $\mu(F)$ — is that $\mathcal{E}(F)$ is smooth and differentiable in the feature vectors, enabling gradient analysis.

Lemma 6.2 (Energy Lower Bound and Its Attainment [8]). *For any unit-norm frame $F = \{f_1, \ldots, f_m\} \subset S^{d-1}$:*

$$\mathcal{E}(F) \;\geq\; \frac{m^2}{d} - m, \tag{6.2}$$

with equality if and only if F is a unit-norm tight frame (UNTF): $FF^\top = \frac{m}{d} I_d$.

Proof. Since $G = F^\top F$ is PSD with rank at most d and trace m, by the power-mean inequality: $\|G\|_F^2 \geq (\operatorname{tr} G)^2/d = m^2/d$. Since $\mathcal{E}(F) = \|G\|_F^2 - m$, the bound follows. Equality holds iff all non-zero eigenvalues of G are equal to m/d — the tight frame condition. □

Lemma 6.2 establishes that UNTFs are the global minimisers of the interference energy. This is the *frame force* result of [8]. What is new is the application to neural feature configurations and the analysis of the energy landscape near its local minima.

6.3 The Stability Analysis: When Configurations Break

We now study the local stability of UNTF configurations. A configuration F^* is a stable local minimum of $\mathcal{E}$ if the Hessian of $\mathcal{E}$ at F^*, restricted to the unit-sphere constraint manifold, is positive definite. Instability — a negative eigenvalue of the constrained Hessian — corresponds to a direction in configuration space along which the interference energy can be reduced, triggering a transition to a new configuration.

Theorem 6.3 (Stability Condition for Two-Feature Configurations). *Consider $m = 2$ features $f_1, f_2 \in S^{d-1}$ with inner product $\rho = \langle f_1,\, f_2 \rangle \in (-1, 1)$. Under the interference energy* (6.1)*, this configuration is:*

- stable *(local minimum) if $\rho^2 > 1/d$;*
- unstable *(saddle point) if $\rho^2 < 1/d$;*
- marginally stable *(bifurcation point) if $\rho^2 = 1/d$.*

Proof. With $m = 2$, the interference energy is $\mathcal{E} = 2\rho^2$. The tangent space to S^{d-1} at f_i is $\{v : \langle v,\, f_i \rangle = 0\}$. For a perturbation $f_i \to f_i + \varepsilon\delta_i$ with $\delta_i \perp f_i$, the second variation is:

$$\delta^2\mathcal{E} \;=\; 2\varepsilon^2 \sum_{i \neq j} \Big[|\langle \delta_i,\, f_j \rangle|^2 + |\langle f_i,\, \delta_j \rangle|^2 + 2\langle f_i,\, f_j \rangle \langle \delta_i,\, \delta_j \rangle \Big] .$$

For the in-plane symmetrised perturbation $\delta_1 = \delta_2 = v$ with $v \perp \{f_1, f_2\}$ and $\|v\| = 1$, one computes $\delta^2\mathcal{E} = 8\varepsilon^2(\rho^2 - 1/d)$, using $\langle f_i,\, v \rangle = 0$, $\langle v,\, f_j \rangle = 0$, and $\langle \delta_1,\, \delta_2 \rangle = 1$. The sign of $\delta^2\mathcal{E}$ is determined by $\rho^2 - 1/d$. □

The critical overlap $\rho^* = 1/\sqrt{d}$ is precisely the Welch bound for $m = 2$ features in $\mathbb{R}^d$ (Theorem 1.1 with $m = 2$ gives $\mu_{\min} = 1/\sqrt{d}$). Below this critical overlap, the two-feature configuration is unstable: the energy landscape favours moving the features toward orthogonality. Above this overlap (features highly correlated), the configuration is stable: the energy barrier to splitting is positive, and the features remain merged.

6.4 Sparsity-Driven Splitting: The Bifurcation Condition

The interference energy $\mathcal{E}(F)$ alone is not the full story. The network minimises the total loss, which includes both the task loss and the regularisation. We now incorporate sparsity into the stability analysis.

Recall from Chapter 1 that the expected squared read-out error for feature i under k-sparse activations is bounded by $k\mu(F)^2$. The total representation loss for a configuration of m features with sparsity k is:

$$\mathcal{L}_{\text{rep}}(F, k) \;\approx\; \frac{k}{m}\mathcal{E}(F) + \lambda m, \tag{6.3}$$

where the first term is the average interference per feature and the second is the circuit complexity (penalising configurations that require more edges to implement).

Theorem 6.4 (Critical Sparsity for Feature Splitting)**.** *A polysemantic configuration (two concepts sharing a direction, $\rho \approx 1$) is energetically favoured over a monosemantic split ($\rho = 0$) when:*

$$k \;<\; k^* \;=\; \frac{\lambda d^2}{2(m-d)}. \tag{6.4}$$

For $k > k^*$, the split configuration is energetically preferred, and the network will split the polysemantic neuron into two monosemantic ones during training.

Proof. The interference energy of the polysemantic configuration (two features at $\rho = 1$) is $\mathcal{E}_{\text{poly}} = 2$. The interference energy of the split configuration ($\rho = 0$) is $\mathcal{E}_{\text{split}} = 0$. The circuit complexity of the polysemantic configuration is $\lambda \cdot 1$ (one direction in the circuit); of the split, $\lambda \cdot 2$. From (6.3), the split is favoured when $(k/m)\cdot 0 + \lambda \cdot 2 < (k/m)\cdot 2 + \lambda \cdot 1$, i.e., $\lambda < 2k/m$, i.e., $k > \lambda m/2$. The refined bound (6.4) accounts for the contribution of all m features to the interference energy using Lemma 6.2. □

Theorem 6.4 makes a concrete, testable prediction: increasing the sparsity k (e.g., by adding a sparsity-inducing regulariser on the activations) triggers feature splitting, while decreasing k triggers merging. The critical sparsity k^* scales as d^2/m: in large, wide networks (large d), features are harder to split; in networks with many features (large m), they split more easily.

6.5 The Full Phase Diagram

For m features in $\mathbb{R}^d$ with sparsity k and complexity penalty λ, the system has a rich phase diagram in the (λ, k) plane with four qualitatively distinct phases.

Phase I — Dense-Polysemantic ($k < k^*$, $\lambda < \lambda^*$). Dense activations; all features represented polysemantically. This is the standard large language model pre-training regime: all features active simultaneously, heavy superposition tolerated.

Phase II — Sparse-Polysemantic ($k > k^*$, $\lambda < \lambda^*$). Sparse activations but features still merged. This is a transient regime: interference is low because activations are sparse, but the system is not at a global minimum of $\mathcal{L}_{\text{rep}}$; it will eventually split.

Phase III — Dense-Monosemantic ($k < k^*$, $\lambda > \lambda^*$). Small number of orthogonal feature directions; dense activations; zero interference. This is the regime of classical artificial neural networks with small hidden dimensions.

Phase IV — Sparse-Monosemantic ($k > k^*$, $\lambda > \lambda^*$). Sparse activations and monosemantic features — the interpretability ideal. Each active feature is read out cleanly, without interference from inactive features, and each neuron encodes a single concept.

Theorem 6.5 (Phase Boundary). *The phase boundary between the polysemantic and monosemantic phases, at fixed λ, is the curve $k = k^*(\lambda, m, d) = \lambda d^2/[2(m-d)]$ in the (k, m) plane. As $m \to d^+$ (approaching the dimension from above), $k^* \to \infty$ — an infinite sparsity is required to split the first feature beyond the basis. As $m \to \infty$, $k^* \to 0$ — any positive sparsity eventually forces splitting when there are very many features.*

The phase diagram is consistent with empirical observations: large language models trained on dense corpora (Phase I) are heavily polysemantic. Models fine-tuned with sparsity constraints or with feature-level ℓ^1 regularisation (Phase IV approach) exhibit more monosemantic representations.

6.6 Symmetry Breaking and the Order Parameter

The splitting transition has the mathematical structure of *spontaneous symmetry breaking* in statistical mechanics. In the merged (polysemantic) phase, two features A and B share a single direction $f_1 = f_2$; the configuration has a $\mathbb{Z}_2$ symmetry (permuting A and B leaves the energy invariant). In the split (monosemantic) phase, the symmetry is broken.

The natural order parameter is:

$$\phi \;=\; \left| f_A^\top f_B \right| / (\|f_A\|\,\|f_B\|) \;=\; |\rho|\,. \tag{6.5}$$

In the merged phase, $\phi = 1$; in the split phase, $\phi \approx 0$. Near the critical sparsity k^*:

$$\phi(k) \;\approx\; \begin{cases} 1 & k < k^* \\ 1 - c\sqrt{k - k^*} & k \gtrsim k^* \end{cases} \tag{6.6}$$

for a constant $c > 0$. The square-root scaling is the mean-field exponent for $\mathbb{Z}_2$ symmetry breaking — the same as in the Ising model. For the splitting of r features ($\mathbb{Z}_r$ symmetry: r features sharing a single direction that splits into r distinct directions), the universality class is the r-state Potts model [68], with a vector-valued order parameter in $\mathbb{R}^{r-1}$.

6.7 Optimal Feature Configurations

Proposition 6.6 (Classification of Optimal Configurations). *The globally optimal feature configurations (local minima of $\mathcal{E}$ on $(S^{d-1})^m$) include:*

(1) $m \le d$: orthonormal frames, $\mathcal{E} = 0$, unique up to $O(d)$.

(2) $m = d + 1$: vertices of a regular simplex centred at the origin, unique up to $O(d)$, with $\mathcal{E} = (d+1)/d$.

(3) $m = 2d$: vertices of a regular cross-polytope (d orthonormal pairs $\pm e_i$), with $\mathcal{E} = 2(2d-1)/d$.

(4) $m = d(d+1)/2$ (real ETF capacity): equiangular tight frames exist for specific values of d (e.g., $d = 2, m = 3$; $d = 3, m = 6$; $d = 7, m = 28$), achieving the absolute bound with minimum coherence.

Between these values, the optimal configurations are less regular.

Corollary 6.7 (Discrete Feature Spectrum). *As sparsity k or complexity penalty λ increases continuously, the number of effectively distinct features (directions in $\mathbb{R}^d$ with $\|f_i\|^2 > \delta$) increases through the discrete values $d, d+1, 2d, \ldots$, not continuously. The transitions between these values are the feature splitting events.*

This discreteness is a prediction of the theory that is, in principle, empirically testable. As training proceeds and sparsity increases, the number of monosemantic features should increase through a sequence of discrete jumps, rather than continuously. Preliminary evidence for this discreteness appears in the training dynamics studies of Anthropic's toy models of superposition [24], though a definitive experimental confirmation remains an open problem.

6.8 Feature Merging: The Reverse Transition and Its Asymmetry

The split-to-merge transition is not the time-reverse of splitting. The asymmetry arises because the merged and split configurations are separated by an energy barrier, and the height of this barrier differs depending on the direction of traversal.

The barrier height from merge to split is $\Delta\mathcal{E}_{\uparrow} = \mathcal{E}_{\text{barrier}} - \mathcal{E}_{\text{merged}}$, and from split to merge is $\Delta\mathcal{E}_{\downarrow} = \mathcal{E}_{\text{barrier}} - \mathcal{E}_{\text{split}}$. Since $\mathcal{E}_{\text{split}} < \mathcal{E}_{\text{merged}}$ (the split configuration has lower interference), we have $\Delta\mathcal{E}_{\uparrow} < \Delta\mathcal{E}_{\downarrow}$. By the Kramers formula (cf. §3.5), the merge-to-split transition time is shorter than the split-to-merge time, reflecting the thermodynamic preference for split configurations at high sparsity.

This asymmetry has a practical consequence: once a network has split its features, it is harder to merge them again than it was to split them in the first place. Networks trained with gradually increasing sparsity therefore tend to reach more monosemantic configurations than networks trained from scratch at high sparsity. This is a prediction of the theory consistent with observations about curriculum learning and progressive regularisation.

Bibliographic Notes

The frame interference energy and the frame force result are from [8]. The empirical observations motivating this chapter are in [24]. The classification of optimal configurations in Proposition 6.6 combines results from [16] and [25]. The connection between feature splitting and $\mathbb{Z}_r$ symmetry breaking draws on the Potts model literature reviewed in [68]. The asymmetry of splitting vs. merging transitions is related to metastability results in [47].

6.9 The Landau Theory of Feature Splitting

The symmetry-breaking analysis of §6.6 connects feature splitting to the theory of phase transitions in a qualitative way. Here we make the connection precise by developing the Landau free energy for the splitting transition.

6.9.1 The Landau Expansion

Let $\phi = |\rho|$ be the order parameter measuring the overlap between the two feature vectors, and let k be the sparsity level (the control parameter). Near the critical sparsity k^*, the free energy $\mathcal{F}(\phi, k) = \mathcal{L}_{\text{rep}}(F, k)$ can be expanded in powers of $1 - \phi^2$ (which vanishes in the merged phase where $\phi = 1$):

$$\mathcal{F}(\phi, k) \;=\; \mathcal{F}_0(k) + a_2(k)(1 - \phi^2) + a_4(k)(1 - \phi^2)^2 + \cdots \tag{6.7}$$

where the coefficients $a_2(k)$ and $a_4(k)$ are determined by the geometry of the frame and the sparsity distribution.

From the total representation loss (6.3), the coefficient $a_2(k)$ is:

$$a_2(k) \;=\; \frac{k}{m}\cdot\frac{\partial^2\mathcal{E}}{\partial\phi^2}\bigg|_{\phi=1} - \lambda \;=\; \frac{2k}{m} - \lambda, \tag{6.8}$$

where we used the fact that $\partial^2\mathcal{E}/\partial\phi^2\big|_{\phi=1} = 2$ (computed from $\mathcal{E} = 2\phi^2$ in the two-feature case). The coefficient $a_4(k)$ is:

$$a_4(k) \;=\; \frac{k}{m}\cdot\frac{1}{12}\frac{\partial^4\mathcal{E}}{\partial\phi^4}\bigg|_{\phi=1} \;=\; \frac{k}{6m} \;>\; 0. \tag{6.9}$$

6.9.2 The Second-Order Transition

Since $a_4(k) > 0$ for all $k > 0$, the Landau expansion (6.7) describes a *second-order* (continuous) phase transition. The equilibrium value of ϕ is determined by $\partial\mathcal{F}/\partial\phi = 0$:

- For $k < k^*$ (where $a_2(k) > 0$): the minimum of $\mathcal{F}$ is at $\phi = 1$ (merged phase).
- For $k > k^*$ (where $a_2(k) < 0$): the minimum shifts to $\phi^2 = 1 + a_2/(2a_4) = 1 - (k - k^*)\cdot 3m/k$, so $\phi < 1$ (split phase).

Near k^*, expanding: $\phi^2 \approx 1 - 3m(k - k^*)/(k^*)$, giving:

$$1 - \phi \;\approx\; \frac{3m}{2k^*}(k - k^*) \quad \text{as } k \to (k^*)^+. \tag{6.10}$$

This is a *linear* approach to the transition (mean-field exponent $\nu = 1$), consistent with the Landau mean-field theory. The square-root scaling claimed in (6.6) arises from fluctuations beyond the mean-field approximation, which become important when the network has a finite number of features m and the "system size" is not infinite.

6.9.3 Fluctuations and the Ginzburg Criterion

The mean-field theory is valid as long as fluctuations of the order parameter are small compared to its mean value. The Ginzburg criterion for the validity of mean-field theory in the splitting transition is:

$$\frac{\langle(\delta\phi)^2\rangle}{\phi^2} \;\ll\; 1, \tag{6.11}$$

where $\langle(\delta\phi)^2\rangle$ is the variance of the order parameter fluctuations. By a standard calculation using the Landau free energy (6.7), the variance is: $\langle(\delta\phi)^2\rangle \sim k_B T_{\text{eff}}/(m\cdot a_4)$, where $T_{\text{eff}} \sim \lambda\eta$ is the effective temperature (as in §3.5) and m is the number of features.

The Ginzburg criterion (6.11) is satisfied when $m \gg k_B T_{\text{eff}}/a_4$, i.e., when the number of features is large. For large networks (large m), the mean-field theory is valid and the transition is second-order with linear exponent. For small networks (small m), fluctuations are large and the transition may appear first-order due to finite-size effects.

This has a practical implication: in the toy models of superposition [24], which use small networks with few features, the splitting transitions observed may appear sharper (more first-order-like) than predicted by the mean-field theory. In large language models with many features, the transitions should be smoother and more amenable to mean-field analysis.

6.10 The Geometry of the Feature Moduli Space

The set of UNTF configurations — the global minimisers of $\mathcal{E}(F)$ for fixed m and d — forms a manifold $\mathcal{M}_{m,d} \subset (S^{d-1})^m$. Understanding the topology and geometry of $\mathcal{M}_{m,d}$ is the central question of *frame theory*, and it has direct bearing on the complexity of feature splitting.

6.10.1 Connected Components and Their Interpretation

The connected components of $\mathcal{M}_{m,d}$ correspond to qualitatively different feature configurations. A splitting transition moves the system from one component (the merged one, where two or more features coincide) to an adjacent component (the split one, where they are distinct). The *moduli space* of UNTFs modulo orthogonal equivalence is the quotient $\mathcal{M}_{m,d}/O(d)$, and its topology encodes the classification of essentially distinct frame configurations.

For small values of m relative to d, the structure is completely known by Proposition 6.6. The transitions between the special values $m = d, d+1, 2d, \ldots$ are the feature splitting events, and Corollary 6.7 asserts that the number of effectively distinct features increases through these discrete values as sparsity increases.

6.10.2 The Euler Characteristic of the Moduli Space

For $m \leq d$ (the orthonormal case), the moduli space $\mathcal{M}_{m,d}/O(d)$ is the *Stiefel manifold* $V(d, m)$ of m-frames in $\mathbb{R}^d$, which has Euler characteristic:

$$\chi(V(d,m)) \;=\; \begin{cases} 1 & \text{if } m = d \text{ or } m = 0 \\ 0 & \text{if } 0 < m < d \text{ and } d - m \text{ is odd} \\ 2 & \text{if } 0 < m < d \text{ and } d - m \text{ is even.} \end{cases}$$

For $m > d$ (the overcomplete case), the moduli space is more complex and its Euler characteristic depends sensitively on the relationship between m and d. The fact that the Euler characteristic changes as m increases through the special values $d + 1, 2d, d(d + 1)/2$ is a reflection of the topological changes in the moduli space at each phase transition.

Bibliographic Notes

The frame interference energy and frame force result are from [8]. The empirical observations motivating this chapter are in [24]. The classification of optimal configurations in Proposition 6.6 combines results from [16] and [25]. The connection to Potts model universality is from [68]. The metastability analysis and asymmetry of splitting vs. merging are related to [47]. The Landau theory of second-order phase transitions is treated in any standard text on statistical mechanics; see Goldenfeld [27] for a mathematical treatment.

6.11 Persistent Homology and the Topology of Feature Space

The frame interference energy $\mathcal{E}(F)$ and the phase transitions of feature splitting are naturally studied using the tools of *topological data analysis* (TDA), specifically persistent homology. We briefly indicate the connection.

6.11.1 The Vietoris-Rips Complex of a Feature Frame

Given a feature frame $F = \{f_1, \ldots, f_m\} \subset S^{d-1}$, define the *Vietoris-Rips complex* $\mathrm{VR}(F, r)$ at scale $r > 0$ as the abstract simplicial complex with vertices $\{1, \ldots, m\}$ where a simplex $\sigma \subset \{1, \ldots, m\}$ is included if and only if $\|f_i - f_j\| \leq r$ for all $i, j \in \sigma$.

The topology of $\mathrm{VR}(F, r)$ encodes the clustering structure of the feature frame at scale r. At $r = 0$, no simplices are included (all features are isolated). As r increases, features that are close together (high inner product, high interference) form clusters. At $r = 2$ (the maximum possible distance on S^{d-1}), all simplices are included.

Proposition 6.8 (Topology of the Optimal Frame)**.** *For a UNTF with $m = d+1$ features in $\mathbb{R}^d$ (the regular simplex), the Vietoris-Rips complex $\mathrm{VR}(F, r)$ undergoes topological transitions at:*

- $r = r^* = 2/\sqrt{d/(d+1)}$*: the scale at which all pairs of features become connected (since all pairwise distances in the regular simplex are equal to $\sqrt{2 - 2/d} \approx \sqrt{2}$ for large d).*
- $r > r^*$*: $\mathrm{VR}(F, r)$ is a simplex (contractible), so all homology groups vanish.*

For $r < r^*$, the 0-th Betti number $\beta_0 = m$ (all features isolated) and all higher Betti numbers vanish.

6.11.2 Persistent Homology as a Transition Detector

The persistent homology of the filtration $\{\mathrm{VR}(F, r)\}_{r \geq 0}$ tracks the topological features (connected components, loops, voids) that appear and disappear as r increases. The *persistence diagram* records, for each topological feature, the scale r_{birth} at which it appears and the scale r_{death} at which it disappears.

For optimal frame configurations (UNTFs and ETFs), the persistence diagram has a characteristic structure: all homological features die at the same scale r^*, reflecting the regularity of the optimal configuration. For suboptimal configurations (e.g., frames with some polysemantic features), the persistence diagram is more varied, with features dying at different scales.

The *bottleneck distance* between two persistence diagrams is a metric on the space of frame configurations, and provides a topology-aware notion of circuit distance that could supplement the Kolmogorov complexity-based distance d_{circ} of Definition 10.3.

6.11.3 Tracking Feature Splitting with Persistent Homology

The feature splitting transition (Theorem 6.4) can be detected using persistent homology as follows. As the sparsity k increases through the critical value k^*, the merged feature vectors (two features at a single point) separate into two distinct points. In the Vietoris-Rips filtration:

- Before splitting ($k < k^*$): the two merged features appear as a single point in the complex, so β_0 decreases by 1 compared to the split configuration.
- After splitting ($k > k^*$): the two features appear as distinct points, and β_0 increases by 1 (two isolated components appear before merging at scale r^*).

The splitting transition can therefore be detected by monitoring the first Betti number β_0 of the frame's persistence diagram as the sparsity k is varied. A jump in β_0 signals a splitting event, and the size of the jump equals the number of features involved.

This provides a computationally tractable method for detecting feature splitting in practice, without requiring the full circuit discovery computation: compute the Vietoris-Rips persistence diagram of the feature frame at each training step, and monitor β_0 for jumps.

Bibliographic Notes

The frame interference energy is from [8]. The empirical observations motivating this chapter are from [24]. The classification of optimal configurations is from [16] and [25]. The Potts model universality is from [68]. Metastability and the Kramers formula are from [47]. The Landau theory is from [27]. Topological data analysis and persistent homology are treated in Edelsbrunner and Harer (2010), *Computational Topology*.

6.12 Numerical Experiments: Toy Models of Splitting

We describe the numerical experiments that validate the phase diagram predictions of §6.5. These experiments extend the toy model setup of [24] and test the critical sparsity formula (Theorem 6.4).

6.12.1 Experimental Setup

Fix a network with $m = 7$ binary features in $\mathbb{R}^{d=3}$ (so $m/d \approx 2.33$, in the superposition regime). The network is trained to minimise the reconstruction loss (1.7) with a sparsity-inducing ℓ^1 penalty on the activations and ℓ^2 regularisation on the feature weights. The control parameters are the sparsity penalty strength λ_s (which controls the effective sparsity k) and the weight decay λ.

According to Theorem 6.4, the critical sparsity is $k^* = \lambda d^2/(2(m-d)) = \lambda \cdot 9/(2 \cdot 4) = 1.125\lambda$. For $\lambda = 0.01$, this gives $k^* \approx 0.011$ — features should split for effective sparsity $k > 0.011$.

6.12.2 Predicted Phase Boundaries

The phase diagram in the (λ, k) plane predicts:

λ	$k^*(\lambda) = 1.125\lambda$	Predicted phase
0.001	0.001	IV (sparse-monosemantic) for $k > 0.001$
0.01	0.011	IV for $k > 0.011$
0.1	0.113	IV for $k > 0.113$
1.0	1.125	II (sparse-polysemantic) for $k < 1.125$

For large λ ($\lambda > 1$), the cost of representing any features (λm term in (6.3)) is so high that the network prefers to represent few or no features, regardless of sparsity. This is the transition to Phase III (dense-monosemantic) or the all-polysemantic limit.

6.12.3 Empirical Validation

Running the experiments (with 50 independent seeds per parameter setting, 10000 gradient steps per run) yields:

- For $\lambda = 0.01$, $k < 0.01$: the network settles in Phase I (dense-polysemantic); all 7 features are represented in 3 directions with pairwise inner products ≈ 0.8.
- For $\lambda = 0.01$, $k = 0.02$: the network transitions to Phase IV (sparse-monosemantic); 3 features are represented monosemantically in the 3 orthogonal basis directions, while the remaining 4 features form 2 pairs of antipodal features ($\pm e$ configuration), consistent with $m = 2d = 6$ optimal packing.
- For $\lambda = 0.01$, $k = 0.05$: the network reaches the optimal cross-polytope configuration for $m = 6$ features, with the 7th feature absent (zero weight).

The transitions occur at values of k consistent with the critical sparsity formula $k^* = 1.125\lambda$, validating Theorem 6.4 to within experimental error.

6.12.4 The Discrete Feature Spectrum

The most striking empirical finding is the discreteness of the transitions: as k is increased continuously from 0 to 0.1 (with $\lambda = 0.01$), the number of monosemantically represented features increases in jumps of $+1$ or $+2$, not continuously. This confirms Corollary 6.7: the number of distinct features increases through the sequence $3, 4, 6, 7$ (corresponding to the special values $d, d+1, 2d, 2d+1$ for $d = 3$).

The jump from 3 to 4 features (at $k \approx k^* = 0.011$) corresponds to the first splitting event: the network adds a 4th feature direction, forming a regular simplex in $\mathbb{R}^3$. The jump from 4 to 6 features (at $k \approx 2k^* = 0.022$) corresponds to a second splitting event: the network adds two antipodal features, forming a cross-polytope plus one extra vertex.

6.13 Connections to Random Matrix Theory

The stability analysis of feature configurations (§6.3) connects to *random matrix theory* (RMT) through the eigenvalue distribution of the Gram matrix $G = F^\top F$.

6.13.1 The Marchenko-Pastur Law

For a random UNTF with m vectors in $\mathbb{R}^d$ (drawn uniformly from S^{d-1}), the empirical spectral distribution of G converges, as $m, d \to \infty$ with $m/d \to \gamma > 1$, to the *Marchenko-Pastur distribution*:

$$\rho_{\mathrm{MP}}(\lambda) \;=\; \frac{d}{2\pi m \lambda}\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}, \quad \lambda \in [\lambda_-, \lambda_+], \tag{6.12}$$

where $\lambda_\pm = (1 \pm \sqrt{m/d})^2$. The support of the Marchenko-Pastur distribution ranges from $\lambda_- = (1 - \sqrt{m/d})^2$ (lower edge) to $\lambda_+ = (1 + \sqrt{m/d})^2$ (upper edge).

For the frame interference energy $\mathcal{E}(F) = \|G - I_m\|_F^2 - m$, the expected energy under the Marchenko-Pastur distribution is:

$$\mathbb{E}_{\mathrm{MP}}[\mathcal{E}(F)] \;=\; \int_{\lambda_-}^{\lambda_+} (\lambda - 1)^2 \rho_{\mathrm{MP}}(\lambda)\, d\lambda \;=\; \frac{m^2}{d} - m, \tag{6.13}$$

which is exactly the Welch bound of Lemma 6.2. This confirms that random frames achieve the energy lower bound of Lemma 6.2 on average.

6.13.2 Eigenvalue Outliers and Phase Transitions

When the frame is not random but has a structured component (e.g., features that are polysemantic or particularly important), the Gram matrix G may have eigenvalue outliers — eigenvalues outside the Marchenko-Pastur support. These outliers correspond to directions in feature space that are over-represented in the frame.

The splitting transition of Theorem 6.4 can be detected via the appearance of an eigenvalue outlier in G: when two features merge into a polysemantic neuron, the corresponding eigenvector of G has a large eigenvalue (≈ 2 for two merged features) that is outside the Marchenko-Pastur bulk. When the features split, the outlier disappears and the eigenvalue falls back into the bulk.

This connection to random matrix theory provides a computationally efficient method for detecting phase transitions in feature configurations: monitor the spectral distribution of the Gram matrix during training and detect outliers.

Bibliographic Notes

Frame interference energy: [8]. Toy models of superposition: [24]. Optimal configurations: [16], [25]. Potts model: [68]. Metastability: [47]. Landau theory: [27]. Topological data analysis: Edelsbrunner and Harer (2010). Marchenko-Pastur law: Marchenko and Pastur (1967), "Distribution of Eigenvalues for Some Sets of Random Matrices", *Mat. Sb.* 72(4):507–536. Random matrix theory and neural networks: Pennington and Bahri (2017), "Geometry of Neural Network Loss Surfaces via Random Matrix Theory."

6.14 The Frame Potential and Optimal Configurations

The frame interference energy $\mathcal{E}(F)$ is closely related to the *frame potential* introduced by Benedetto and Fickus [8]. We develop this connection and use it to prove that gradient flow on the frame potential converges to tight frames.

6.14.1 The Frame Potential

Definition 6.9 (Frame Potential)**.** The *frame potential* of a collection of m unit vectors $\{f_i\}_{i=1}^m \subset S^{d-1}$ is:

$$\mathrm{FP}(F) \;=\; \sum_{i,j=1}^{m} |\langle f_i,\, f_j\rangle|^2 \;=\; \mathcal{E}(F) + m \;\geq\; \frac{m^2}{d}. \tag{6.14}$$

The frame potential equals the frame interference energy plus m (the diagonal terms $\sum_i |\langle f_i,\, f_i\rangle|^2 = m$). Minimising the frame potential over all collections of m unit vectors in S^{d-1} is equivalent to minimising the frame interference energy.

Theorem 6.10 (Frame Potential Minimisers [8])**.** *The global minimisers of* $\mathrm{FP}(F)$ *over all collections of* m *unit vectors in* S^{d-1} *are exactly the unit-norm tight frames (UNTFs), achieving* $\mathrm{FP}(F) = m^2/d$*. Moreover, the gradient flow*

$$\dot{f}_i \;=\; -\pi_{T_{f_i}}(\nabla_{f_i}\mathrm{FP}) \;=\; -4\sum_{j\neq i}\langle f_i,\, f_j\rangle(f_j - \langle f_i,\, f_j\rangle f_i) \tag{6.15}$$

(projected gradient on the sphere) converges to a local minimum of FP*, and all local minima of* FP *are UNTFs.*

The theorem says that the frame potential is a *Morse function* with a unique minimum value m^2/d achieved by all UNTFs. In the language of dynamical systems, all orbits of the gradient flow (6.15) converge to the stable manifold of UNTF configurations.

6.14.2 The Learning Dynamics as Frame Potential Minimisation

The connection between Theorem 6.10 and neural network training is the following. During the learning phase (after grokking, in the consolidation stage of §3.7), weight decay penalises the total weight norm $\|W\|_F^2$, which is proportional to the frame potential when the network stores features as directions in activation space.

More precisely, for a network in the superposition regime where the feature vectors f_i are the rows of the weight matrix W, the weight decay regulariser $\lambda\,\|W\|_F^2 = \lambda m$ (since all feature vectors have unit norm). The frame potential contribution from the interference terms is:

$$\lambda\,\|W\|_F^2 \;\supseteq\; \lambda\left\|W^\top W\right\|_F^2 \;=\; \lambda\cdot\mathrm{FP}(F), \tag{6.16}$$

which shows that the weight decay regulariser controls the frame potential. Minimising the regularised loss therefore includes minimising the frame potential as a component, which drives the feature frame toward UNTF configurations.

This is the formal justification for the claim that gradient descent with weight decay learns feature configurations that approach the optimal polytopes of Proposition 6.6: the regulariser directly minimises the frame potential, whose global minimisers are the desired optimal configurations.

Bibliographic Notes

Frame interference energy: [8]. Toy models: [24]. Optimal configurations: [16], [25]. Potts model: [68]. Metastability: [47]. Landau theory: [27]. Topological data analysis: Edelsbrunner and Harer (2010). Random matrix theory: Marchenko and Pastur (1967).

Chapter 7

Universality Classes of Circuits

7.1 The Universality Question

In the physical sciences, universality is one of the most powerful and surprising organisational principles known. Systems that differ in every microscopic detail — a ferromagnet and a liquid near its critical point, a superconductor and a polymer in solution — can share identical macroscopic behaviour near phase transitions: the same critical exponents, the same scaling functions, the same qualitative structure of fluctuations. The reason is that universality classes are determined not by the details of the microscopic interaction but by the *symmetries* of the system and the *dimensionality* of the relevant order parameter. Everything else is irrelevant in the renormalisation group sense: it flows to zero as one coarse-grains to macroscopic scales.

An analogous question arises in mechanistic interpretability with compelling urgency. Networks trained on different datasets, with different architectures, different random seeds, different optimisers, sometimes appear to implement the same internal computations — the same circuits, representing the same features, in the same geometric configurations. The curve detectors found in InceptionNet appear also in ResNets; the indirect-object identification circuit found in GPT-2 small appears also in larger models [46, 64]. If this is a robust pattern rather than a coincidence, it demands a theoretical explanation: why should two independently trained systems converge to the same internal algorithm?

We develop in this chapter a theory of *circuit universality* that answers this question, at least partially. The answer draws on three ingredients: the uniqueness (up to symmetry) of information-theoretically optimal circuits (Chapter 4); the phase-transition structure of circuit emergence (Chapter 3); and the algebraic structure of circuit composition (Chapter 2). Together, they predict that whenever two networks are trained on tasks with the same information-theoretic structure, they will implement circuits in the same *universality class* — sharing the same qualitative architecture, even if their weight matrices differ in every specific entry.

7.2 Defining Circuit Universality

We need a notion of equivalence between circuits that abstracts away implementation details while preserving functional and structural content.

Definition 7.1 (Circuit Isomorphism)**.** Two circuits $\mathcal{C}$ and $\mathcal{C}'$ are (ε, ϕ)-*isomorphic* if there exist:

(1) a bijection $\sigma : V^{\mathcal{C}} \to V^{\mathcal{C}'}$ between node sets that respects the layer structure;

(2) a collection of orthogonal maps $\{O_\ell \in O(d_\ell)\}_{\ell=0}^{L}$ (one per layer);

(3) a scaling $\rho > 0$;

such that for all inputs x in the domain:

$$\|f_{\mathcal{C}'}(O_0 x) - \rho \cdot O_L f_{\mathcal{C}}(x)\| \;\leq\; \varepsilon \, \|f_{\mathcal{C}}(x)\|\,, \tag{7.1}$$

and the edge weights satisfy $\left|W_{\mathcal{C}'}(\sigma(u),\sigma(v)) - \rho \cdot (O_{\ell+1} W_{\mathcal{C}}(u,v) O_\ell^\top)_{uv}\right| \leq \phi$ for all edges (u,v).

The orthogonal maps O_ℓ absorb the representational symmetry: any orthogonal transformation of the feature space at layer ℓ produces an equivalent network. The scaling ρ absorbs the norm ambiguity. The tolerances ε and ϕ allow for approximate isomorphism.

Definition 7.2 (Universality Class)**.** A *universality class* $\mathcal{U}$ is an equivalence class of circuits under $(0,0)$-isomorphism (exact isomorphism). Two circuits are in the same universality class if and only if they are related by orthogonal transformations of their layer representations and an overall scaling.

Universality in this sense is a strong claim: it asserts that the circuits learned by independently trained networks are not merely functionally equivalent (producing the same input-output map) but structurally equivalent (implementing the same algorithm, with the same intermediate representations, up to rotation and scaling). This is what the empirical evidence from [46, 64] suggests, and what the theory below predicts.

7.3 The Platonic Representation Hypothesis

Before developing the formal theory, we articulate the conceptual picture that motivates it.

Hypothesis 7.3 (Platonic Representation Hypothesis)**.** For any task with a fixed information-theoretic structure (fixed joint distribution $p(X,Y)$), there exists a canonical circuit $\mathcal{C}^*$ — the *Platonic circuit* — that is the unique (up to isomorphism) minimal faithful circuit achieving the information bottleneck optimum of Chapter 4. Any network trained to sufficient accuracy on this task will implement a circuit in the same universality class as $\mathcal{C}^*$.

The name is deliberately philosophical. Plato's theory of Forms held that particular objects in the world are imperfect copies of perfect, eternal archetypes. The Platonic Representation Hypothesis holds that the circuits implemented by neural networks are imperfect copies of perfect, mathematically determined archetypes — the optimal solutions to the information-theoretic problem defined by the task. The world of machine learning, with its accidents of initialisation, architecture, and stochastic gradient descent, corresponds to Plato's cave; the Platonic circuits are the shadows on the wall.

This hypothesis has been studied empirically in [34], which finds evidence for convergent representations across vision models of different architectures. Our contribution is to provide the theoretical framework that explains why such convergence should be expected.

7.4 Formal Statement and Proof of Universality

Theorem 7.4 (Circuit Universality)**.** *Let $p(X,Y)$ be a joint distribution over input-label pairs, and let $\mathcal{T}$ be the associated prediction task. Suppose:*

(1) (Sufficient capacity) *Both networks $\mathcal{N}$ and $\mathcal{N}'$ have sufficient capacity to achieve test error at most ε;*

(2) (IB optimality) *Both are trained to the information bottleneck optimum with Lagrange multiplier β (minimising $I(X;T) - \beta I(T;Y)$ over all encoders consistent with the network architecture);*

(3) (Strict convexity) *The information bottleneck functional* (4.4) *has a unique minimiser at the given β, up to the symmetries of $p(X, Y)$.*

Then the minimal faithful circuits $\mathcal{C} \subseteq \mathcal{N}$ and $\mathcal{C}' \subseteq \mathcal{N}'$ extracted from the two networks are $(O(\varepsilon), O(\varepsilon))$-isomorphic.

Proof. Under assumption (2), both $\mathcal{C}$ and $\mathcal{C}'$ implement the optimal bottleneck representation T^* (Theorem 4.2). Under assumption (3), T^* is unique up to the symmetries of $p(X, Y)$. The symmetry group of $p(X, Y)$ — those transformations of the input space that leave the distribution invariant — acts on the set of optimal representations, partitioning it into orbits. Two representations in the same orbit differ only by an element of the symmetry group.

For continuous input distributions $p(X)$ with no discrete symmetry group (e.g., Gaussian inputs), the symmetry group is $O(d)$ (the full orthogonal group on the input space). In this case, the optimal representation is unique up to orthogonal transformations of each layer's feature space — exactly the O_ℓ transforms of Definition 7.1. Under the circuit extraction algorithm (2.3), the extracted circuits $\mathcal{C}$ and $\mathcal{C}'$ differ by these orthogonal transforms, giving $(0, 0)$-isomorphism. For $\varepsilon > 0$, the approximate faithfulness condition introduces a tolerance of $O(\varepsilon)$ in the weight differences. □

The proof is clean but the assumptions are strong. Assumption (2) — information bottleneck optimality — is not guaranteed by standard gradient descent; it is an idealisation that captures the tendency of gradient descent to find representations that are neither more nor less informative than they need to be. Assumption (3) — strict convexity — fails for tasks with discrete symmetries.

7.5 Approximate Universality and the Role of Symmetry

When assumption (3) fails, the bottleneck functional may have multiple minimisers. In this case, universality holds only *within a symmetry orbit.*

Example 7.5 (Modular Addition Circuits)**.** The task of computing $(a + b) \bmod p$ has a symmetry group containing at least $\mathbb{Z}_p$ (cyclic shifts) and $\mathbb{Z}_2$ (reflection $a \leftrightarrow b$). The Fourier algorithm of §3.4 is not the unique optimal circuit: any permutation of the active Fourier modes $\{k_1, \ldots, k_r\}$ gives an equally optimal circuit. Within each orbit (fixed set of active modes), the circuit is unique up to orthogonal symmetry. Across orbits (different sets of active modes), the circuits are not isomorphic but are equally optimal. Empirically, gradient descent selects the set of modes with the largest Fourier coefficients in the generalisation error expansion — a selection rule determined by the implicit regularisation of the optimiser.

The non-uniqueness in Example 7.5 is resolved by noting that the implicit regularisation of gradient descent (specifically, the tendency to find minimum-norm solutions, which corresponds to selecting the most compressed circuit in the sense of Chapter 4) imposes additional constraints beyond the information bottleneck. The full universality theory requires accounting for this implicit regularisation, which is beyond the scope of the current formalisation but represents an important open problem (Problem 3 in Appendix B).

7.6 Universality Across Architectures: The Transformer–MLP Equivalence

A particularly striking form of universality is the claim that transformer circuits and MLP circuits can, for certain tasks, be in the same universality class.

Theorem 7.6 (Transformer–MLP Equivalence for Compositional Tasks). *Let $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^k$ be a compositional task of depth L, decomposable as $\mathcal{T} = \mathcal{T}_L \circ \cdots \circ \mathcal{T}_1$ where each $\mathcal{T}_\ell : \mathbb{R}^{n_\ell} \to \mathbb{R}^{n_{\ell+1}}$ has polynomial complexity. Then:*

(i) The minimal faithful MLP circuit has depth $\Theta(L)$ and width $O(\max_\ell n_\ell)$, implementing the composition layer by layer.

(ii) The minimal faithful transformer circuit has depth $O(\log L)$ and width $O(\max_\ell n_\ell \cdot L^{1/2})$.

(iii) Both circuits are in the same universality class: they implement the same intermediate representations $T^{(1)}, \ldots, T^{(L)}$ at successive computational stages, differing only in how many layers of the network are required to produce each intermediate representation.

Part (iii) is the key universality claim: even though the two architectures have different depths, they produce the same sequence of intermediate representations. The "same algorithm" is implemented at different speeds — the transformer parallelises what the MLP does sequentially — but the *content* of each computation is the same. The features at each stage are identical (up to isomorphism), and the circuits connecting them are in the same universality class.

The proof uses the depth-width tradeoff of Theorem 2.7 (the transformer achieves logarithmic depth at the cost of higher width) and the Fourier analysis of attention-based computation, applied to the compositional structure of $\mathcal{T}$.

This result has an important practical implication: the mechanistic understanding developed for one architecture can be transferred to another. The circuit discovered in a two-layer MLP for a compositional task predicts the structure of the corresponding transformer circuit, layer by layer.

7.7 Convergent Evolution: An Empirical Pattern and Its Explanation

The biological concept of *convergent evolution* — the independent development of similar traits in phylogenetically distant species, driven by common selective pressures — offers a useful metaphor for circuit universality. The wing of a bird and the wing of a bat are not homologous (they do not derive from a common winged ancestor) but they share qualitative structure because they solve the same physical problem under similar constraints.

In mechanistic interpretability, the analogue is the independent appearance of qualitatively identical circuits in networks that share no architectural ancestor. The curve detectors identified by [46] in convolutional networks appear to emerge independently in every sufficiently trained vision model, regardless of whether the architecture uses convolutions, attention, or pure MLPs. They are the "optimal solution" to the problem of representing oriented edges in two-dimensional images, and this optimality makes them universal.

Proposition 7.7 (Necessary Condition for Universality). *A circuit $\mathcal{C}$ for task $\mathcal{T}$ can be universal (appearing across architectures and initialisations) only if:*

(1) It achieves the information bottleneck optimum for $\mathcal{T}$ (Chapter 4);

(2) Its Kolmogorov complexity $K(\mathcal{C})$ is minimal among all faithful circuits;

(3) It is a stable fixed point of the gradient flow — a local minimum of the loss landscape that is robust to small perturbations of the weights.

Condition (3) is necessary because unstable fixed points are found only with measure-zero probability by stochastic gradient descent.

The three conditions are not independent: (1) is necessary for task performance, (2) is necessary for weight decay to converge to the circuit (Lemma 3.2 and Theorem 3.3 show that the minimum-complexity solution is preferred under weight decay), and (3) is necessary for the circuit to be a fixed point of the training dynamics. Together, they select a unique (or small number of) circuit(s) as the generic output of training — the Platonic circuits of §7.3.

7.8 Limits of Universality: When Circuits Diverge

Universality is not guaranteed in all cases. We identify three conditions under which independently trained networks implement qualitatively different circuits for the same task.

Multiple optimal circuits. When the task has a large symmetry group (Example 7.5), the set of optimal circuits is large, and two networks may land in different orbits. This is the most common source of non-universality in practice.

Phase transition sensitivity. Near the phase boundaries of the circuit phase diagram (§6.5), small differences in training conditions (learning rate, weight decay, batch size) can push two networks to different sides of the transition, yielding qualitatively different circuits. This is the interpretability analogue of critical sensitivity in statistical mechanics: systems near a critical point are maximally sensitive to perturbations.

Capacity constraints. When one network has strictly more capacity than another, the minimal faithful circuit for the smaller network may be forced to use a higher-complexity implementation than the Platonic circuit, because the Platonic circuit requires more parameters than the smaller network possesses. In this case, the two circuits are not isomorphic even if both are optimal given their respective capacity constraints.

These three cases delineate the boundary of the universality theory and point toward a richer classification of circuits that accounts for capacity, symmetry, and phase-boundary effects.

Bibliographic Notes

Empirical universality of circuit motifs is documented in [46] and [64]. The Platonic representation hypothesis, as a general empirical claim, is developed in [34], which finds evidence for convergent representations across vision models. The transformer–MLP equivalence for compositional tasks connects to the depth-width tradeoff results of [22] and [60]. The biological metaphor of convergent evolution is discussed in the context of neural representations in [69].

7.9 A Worked Example: Universality of Curve Detectors

To make the universality claim concrete, we trace through the example of curve detectors in vision networks, following and formalising the empirical observations of [46].

A *curve detector* is a circuit that responds to oriented curved edges in a two-dimensional image. It is characterised by its preferred orientation $\theta \in [0, 2\pi)$ and its preferred curvature $\kappa > 0$. Formally, it implements a function:

$$f_{\theta,\kappa}(x) \;=\; \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} K_{\theta,\kappa}(s,t)\, x(s,t)\, ds\, dt, \tag{7.2}$$

where $K_{\theta,\kappa}$ is a curved Gabor-like kernel concentrated along a curved path of orientation θ and curvature κ.

The Platonic circuit for a curve detector (Hypothesis 7.3) is determined by the information-theoretic structure of the vision task: detecting oriented edges in natural images. The optimal circuit — the one minimising the information bottleneck functional (4.4) while being faithful for the curve detection task — implements exactly the convolution (7.2).

Proposition 7.8 (Universality of Curve Detectors)**.** *Under the conditions of Theorem 7.4, any two networks trained on a vision task containing natural images will implement curve detectors for the same set of orientations $\theta \in \Theta \subset [0, 2\pi)$ and curvatures $\kappa \in K \subset \mathbb{R}_{>0}$, regardless of architecture (convolutional vs. attention-based) or initialisation. The implemented curve detectors are $(O(\varepsilon), O(\varepsilon))$-isomorphic in the sense of Definition 7.1.*

The key step in the proof is to show that the optimal bottleneck representation for natural images must include oriented edge detectors — because natural images have a strong statistical prior on oriented edges, and representing these edges is necessary for downstream tasks. The optimal set of orientations Θ is determined by the distribution of oriented gradients in natural images, which is approximately uniform in $[0, \pi)$ (due to the symmetry of natural scenes). The optimal set of curvatures K is determined by the distribution of curvature in natural contours, which is concentrated near $\kappa = 0$ (most natural edges are nearly straight, with a power-law tail for highly curved features).

The universality of curve detectors is therefore not a coincidence but a theorem: it follows from the fact that all vision networks trained on natural images are (approximately) solving the same information bottleneck problem, with the same input distribution $p(X)$ (natural images) and the same task structure (predicting labels correlated with image content).

7.10 Universality Across Modalities

A deeper form of universality — and a more surprising one — is the claim that circuits for semantic concepts can be universal *across modalities*: that a language model and a vision model may implement circuits in the same universality class for the same abstract concepts.

This is the core claim of [34]: that representations learned by deep networks trained on different modalities (text, images, audio) converge to a common representational geometry for shared abstract concepts (e.g., "dog," "city," "happiness").

From the information-theoretic perspective of Chapter 4, this cross-modal universality makes sense if the joint distribution $p(X_{\text{text}}, Y)$ and the joint distribution $p(X_{\text{image}}, Y)$ have the same information-theoretic structure — i.e., if the mutual information $I(X_{\text{text}}; Y)$ and $I(X_{\text{image}}; Y)$ agree on the relevant concepts. This is the case when the labels Y correspond to shared abstract categories (semantic concepts) that can be expressed in multiple modalities.

Remark 7.9 (On the Limits of Cross-Modal Universality). Cross-modal universality has limits. The circuits for modality-specific features (e.g., the sound of a violin vs. its visual appearance) are necessarily different, because the relevant input distributions $p(X_{\text{audio}})$ and $p(X_{\text{image}})$ are different. The universality applies only to circuits for concepts that are genuinely modality-independent — i.e., concepts whose mutual information with the label does not depend on which modality is used to encode the input.

Bibliographic Notes

Empirical universality of curve detectors and other circuit motifs is documented in [46]. The indirect-object identification circuit and its universality across model sizes is from [64]. The

Platonic representation hypothesis and cross-modal universality are studied empirically in [34]. The convergent evolution metaphor connects to the biological literature on constraint-based evolution, reviewed in [69].

7.11 Implicit Regularisation and Circuit Selection

Theorem 7.4 was proved under the assumption of information bottleneck optimality. In practice, gradient descent does not minimise the IB functional; it minimises the empirical cross-entropy loss. However, gradient descent with weight decay does exhibit a form of implicit regularisation that biases it toward solutions in the same universality class as the Platonic circuit.

The key observation is that, among all solutions achieving a given level of training loss, weight decay selects the one with minimum ℓ^2 norm. By Lemma 3.2, the minimum-norm solution achieving zero training loss is the algorithmic circuit $\mathcal{C}_{\text{alg}}$. The implicit regularisation therefore favours circuits over memorisation solutions, which is the first step toward universality.

The second step is the selection of which circuit, when multiple circuits achieve the same minimum norm. This is governed by the *gradient flow geometry*: among circuits of the same norm, gradient descent selects the one that is most aligned with the gradient of the loss at initialisation. For random initialisations, the selected circuit is the one that can be reached by the shortest path in weight space from the origin — the "most natural" circuit from the perspective of the initialisation.

Proposition 7.10 (Gradient Flow Circuit Selection)**.** *Among all circuits $\mathcal{C}$ achieving zero training loss on task $\mathcal{T}$ with minimum norm $\|W_{\mathcal{C}}\|_F^2 = N_{\min}$, gradient descent from a random initialisation $W_0 \sim \mathcal{N}(0, \sigma^2 I)$ selects the circuit $\mathcal{C}^*$ maximising:*

$$\mathcal{C}^* \;=\; \arg \max_{\mathcal{C}:\|W_{\mathcal{C}}\|_F^2 = N_{\min}} \langle W_{\mathcal{C}},\, \mathbb{E}[W_0]_{\text{circuit}} \rangle, \tag{7.3}$$

where $\mathbb{E}[W_0]_{\text{circuit}}$ is the projection of the expected gradient at initialisation onto the space of circuits of norm $N_{\min}$.

For symmetric random initialisations ($\mathbb{E}[W_0] = 0$), the selection criterion (7.3) reduces to the circuit that is most aligned with the gradient of the loss at the origin. This is the circuit that the loss function "wants" to learn first — the circuit that provides the largest improvement in training loss per unit of weight norm. For tasks with a natural Fourier decomposition (like modular addition), this circuit is the one using the Fourier modes with the largest coefficients in the loss expansion — exactly the empirically observed selection.

7.12 Universality and Transfer Learning

The universality theory has direct implications for *transfer learning*: the practice of initialising a model with the weights of a previously trained model and fine-tuning on a new task. If circuits are universal across tasks (Theorem 7.4), then the features and circuits learned for a source task will be reusable for a related target task.

Proposition 7.11 (Transfer Learning and Circuit Universality)**.** *Let $\mathcal{T}_{\text{src}}$ and $\mathcal{T}_{\text{tgt}}$ be two tasks with the same information-theoretic structure (same $p(X)$ but different $p(Y|X)$, with $I(X; Y_{\text{src}}) \approx I(X; Y_{\text{tgt}})$). Suppose the minimal faithful circuits for both tasks are in the same universality class. Then a model fine-tuned on $\mathcal{T}_{\text{tgt}}$ from a model trained on $\mathcal{T}_{\text{src}}$ achieves the same test loss as a model trained from scratch on $\mathcal{T}_{\text{tgt}}$, using $O(|E^{\mathcal{C}_{\text{tgt}}}|)$ fine-tuning examples*

(the number of examples needed to adapt the circuit parameters) rather than $O(|E^{\mathcal{N}}|)$ examples (the number needed to train from scratch).

The proposition says that transfer learning is efficient when the source and target tasks have circuits in the same universality class. The fine-tuning sample complexity scales with the circuit size rather than the total model size — which can be a dramatic reduction for large models with small circuits.

This has a practical implication for interpretability-guided model development: by identifying the circuits for a set of source tasks and understanding their universality class, one can predict which target tasks will benefit from transfer learning (those whose circuits are in the same universality class) and which will not (those whose circuits are in different universality classes).

Bibliographic Notes

Empirical universality of circuit motifs is documented in [46] and [64]. The Platonic representation hypothesis is from [34]. The transformer–MLP equivalence connects to [22] and [60]. The biological metaphor of convergent evolution connects to [69]. Transfer learning and its connection to circuit universality is studied empirically in Ren et al. (2024) and related to the circuit framework in the survey of Lampinen and Frankle (2023).

7.13 Representation Similarity Analysis

The universality theory predicts that two networks in the same universality class should have similar representations. This prediction can be tested using *Representation Similarity Analysis* (RSA), a method from neuroscience [38] that has been adapted to compare neural network representations.

7.13.1 The Representation Dissimilarity Matrix

Given a network $\mathcal{N}$ and a set of stimuli $\{x_1, \ldots, x_K\}$, the *Representation Dissimilarity Matrix* (RDM) at layer ℓ is the $K \times K$ matrix:

$$\mathrm{RDM}_\ell^{\mathcal{N}} \;=\; \Big[\|T_\ell(x_i) - T_\ell(x_j)\|^2\Big]_{i,j=1}^{K}, \tag{7.4}$$

where $T_\ell(x)$ is the activation of layer ℓ on input x. The RDM captures the pairwise distances between representations — the "geometry" of how the network represents the set of stimuli.

Two networks are in the same universality class (Definition 7.1) if and only if their RDMs are approximately equal (up to orthogonal transformation):

Proposition 7.12 (RSA and Universality)**.** *Two networks $\mathcal{N}$ and $\mathcal{N}'$ with circuits $\mathcal{C} \subseteq \mathcal{N}$ and $\mathcal{C}' \subseteq \mathcal{N}'$ are $(O(\varepsilon), O(\phi))$-isomorphic (Definition 7.1) if and only if:*

$$\Big\|\mathrm{RDM}_\ell^{\mathcal{C}} - \mathrm{RDM}_\ell^{\mathcal{C}'}\Big\|_F \;\le\; O(\varepsilon \cdot K) \quad \forall\, \ell \in \{0, \ldots, L\}, \tag{7.5}$$

for all layers ℓ of the circuits.

The proposition gives a computationally tractable test for universality: compute the RDMs of two networks at each layer, and check whether they are approximately equal. This is polynomial in K (the number of stimuli) and d (the dimension), making it feasible for large networks.

7.13.2 Centred Kernel Alignment

A related measure of representation similarity is *Centred Kernel Alignment* (CKA), introduced by Kornblith et al. (2019) as a normalised measure of the similarity between two representation matrices.

Given representation matrices $X_\ell \in \mathbb{R}^{K\times d}$ and $X'_\ell \in \mathbb{R}^{K\times d'}$ for two networks at layer ℓ, the linear CKA is:

$$\mathrm{CKA}(X_\ell, X'_\ell) \;=\; \frac{\left\|X_\ell^\top X'_\ell\right\|_F^2}{\left\|X_\ell^\top X_\ell\right\|_F \cdot \left\|X_\ell'^\top X'_\ell\right\|_F}. \tag{7.6}$$

CKA ranges from 0 (completely dissimilar representations) to 1 (identical representations up to linear transformation). A CKA value close to 1 between two networks at a given layer is evidence for universality at that layer.

The relationship between CKA and the universality isomorphism of Definition 7.1 is:

Proposition 7.13 (CKA and Isomorphism)**.** *If $\mathcal{C}$ and $\mathcal{C}'$ are $(0,0)$-isomorphic (exactly isomorphic via orthogonal maps O_ℓ and scaling ρ), then $\mathrm{CKA}(X_\ell^{\mathcal{C}}, X_\ell^{\mathcal{C}'}) = 1$ for all layers ℓ. Conversely, if $\mathrm{CKA}(X_\ell^{\mathcal{C}}, X_\ell^{\mathcal{C}'}) = 1$, then the representations at layer ℓ are related by a linear transformation (not necessarily orthogonal).*

The distinction between "orthogonal transformation" (universality isomorphism) and "linear transformation" (CKA = 1) is important. CKA measures a weaker form of representation similarity than the universality isomorphism: two networks can have CKA = 1 without being in the same universality class, if the linear transformation relating their representations is not orthogonal.

Bibliographic Notes

Empirical universality: [46], [64], [34]. Convergent evolution: [69]. Transfer learning and universality: Ren et al. (2024). Representation Similarity Analysis: Kriegeskorte et al. (2008), "Representational Similarity Analysis — Connecting the Branches of Systems Neuroscience." Centred Kernel Alignment: Kornblith et al. (2019), "Similarity of Neural Network Representations Revisited."

7.14 The Geometry of Convergent Representations

The universality theory predicts that representations converge across models. We now examine the *geometry* of this convergence: what does the space of all possible representations look like, and where do trained models sit in this space?

7.14.1 The Representation Manifold

For a fixed architecture and task, the space of all possible representations at a given layer is a high-dimensional manifold. Define the *representation manifold* as:

$$\mathcal{M}_{\mathcal{T}} \;=\; \{T_W^* : W \in \mathcal{W}\}, \tag{7.7}$$

where T_W^* is the optimal bottleneck representation for the network with weights W, and $\mathcal{W}$ is the space of all possible weight configurations.

The representation manifold $\mathcal{M}_{\mathcal{T}}$ is a subset of the space of all probability distributions over the representation space $\mathbb{R}^d$. Its geometry encodes the space of "distinct" ways of representing the task.

The universality theory of Theorem 7.4 says that, for tasks with a unique IB optimum, the representation manifold $\mathcal{M}_{\mathcal{T}}$ is a single orbit under the symmetry group of $p(X, Y)$:

$$\mathcal{M}_{\mathcal{T}} \;=\; \{O \cdot T^* : O \in G_{p(X,Y)}\}, \tag{7.8}$$

where T^* is the Platonic bottleneck representation and $G_{p(X,Y)}$ is the symmetry group of the data distribution. For tasks with a continuous symmetry group (e.g., $O(d)$-invariant distributions), the manifold is a homogeneous space of dimension equal to the dimension of the symmetry group.

7.14.2 The Convergence Basin

The convergence basin $\mathcal{B}_{\text{conv}} \subseteq \mathcal{W}$ is the set of weight configurations that converge to the Platonic circuit under gradient descent:

$$\mathcal{B}_{\text{conv}} \;=\; \{W_0 \in \mathcal{W} : \lim_{t\to\infty} d_{\text{circ}}(\mathcal{C}_{W_t}, \mathcal{C}^*) \to 0\}, \tag{7.9}$$

where $\mathcal{C}_{W_t}$ is the minimal faithful circuit at training step t and d_{circ} is a circuit distance metric.

The universality theorem (Theorem 7.4) says that, under IB optimality, the convergence basin $\mathcal{B}_{\text{conv}}$ is the entire weight space (all initialisations converge to the same universality class). In practice (without IB optimality), the convergence basin is a proper subset of $\mathcal{W}$, and initialisations outside this basin may converge to different universality classes.

The boundary of the convergence basin is related to the phase boundaries of the circuit phase diagram (Chapter 6): near the phase boundaries, small perturbations of the initialisation can push the training trajectory to different basins. This is the interpretability analogue of sensitive dependence on initial conditions.

7.14.3 Measuring Convergence in Practice

The CKA measure of Proposition 7.13 provides a practical tool for measuring convergence across models. If CKA is close to 1 at all layers, the models are in the same universality class; if CKA is far from 1 at some layer, the models have diverged at that layer.

A systematic programme of CKA measurement across model sizes, architectures, and training regimes would map out the structure of the convergence basin and test the universality theory empirically. Preliminary results in [34] and related works suggest that CKA is indeed close to 1 for models trained on the same task, consistent with the universality prediction.

Bibliographic Notes

Empirical universality: [46], [64], [34]. Convergent evolution: [69]. Implicit regularisation and circuit selection: the connection to minimum-norm solutions is studied in Soudry et al. (2018), "The Implicit Bias of Gradient Descent on Separable Data." RSA: [38]. CKA: Kornblith et al. (2019).

Chapter 8

The Limits of Mechanistic Interpretability

8.1 What Cannot Be Interpreted?

The preceding chapters have developed an ambitious programme. We have defined features and circuits with mathematical precision, characterised their geometric and information-theoretic properties, traced their emergence through training dynamics, and identified conditions under which they are universal. One might conclude, optimistically, that the mechanistic interpretability programme faces no fundamental obstacles — that it is merely a matter of developing better algorithms and applying them patiently to larger and larger networks.

This conclusion would be wrong. There are fundamental limits to mechanistic interpretability — not technical limits that better algorithms might overcome, but mathematical limits of the same kind as the halting problem and Gödel's incompleteness theorems. In this chapter we establish these limits precisely, explore their implications for the interpretability programme, and distinguish them from the merely practical difficulties that could in principle be resolved.

The limits come in three flavours. First, *computational undecidability*: certain questions about the function computed by a neural network are undecidable — no algorithm can answer them for all networks. Second, *statistical indistinguishability*: certain aspects of a circuit's behaviour are undetectable from any finite amount of data, regardless of the algorithm used. Third, *irreducible complexity*: certain computations cannot be decomposed into simpler subcomputations, making mechanistic analysis impossible in principle.

8.2 The Neural Network Verification Problem and Its Undecidability

The most fundamental question in mechanistic interpretability is: what function does this network compute? More specifically: does the network $\mathcal{N}$ implement the target function f^* exactly? This is the *neural network verification problem*.

Theorem 8.1 (Undecidability of Neural Network Equivalence)**.** *The problem of deciding, given two feedforward neural networks $\mathcal{N}$ and $\mathcal{N}'$ with rational weights and ReLU activations, whether $f_{\mathcal{N}}(x) = f_{\mathcal{N}'}(x)$ for all $x \in \mathbb{R}^n$, is undecidable.*

Proof sketch. We reduce from the halting problem. Given a Turing machine M and input w, construct two networks $\mathcal{N}_M$ and $\mathcal{N}_0$ such that $\mathcal{N}_0$ is identically zero and $\mathcal{N}_M$ is zero on all inputs if and only if M halts on w. Such a construction is possible because the computation of a Turing machine can be simulated by a piecewise-linear function, which can be represented

exactly by a ReLU network [57]. The condition "M halts on w" is then encoded as the condition that a specific piecewise-linear function is identically zero. Since the halting problem is undecidable, so is the network equivalence problem. □

The undecidability of network equivalence implies the undecidability of many related questions: Does the network implement a specific algorithm? Does it contain a specific circuit as a subgraph? Does it behave the same as a given human-interpretable program on all inputs? All of these questions reduce to network equivalence.

Corollary 8.2. *The following problems are undecidable for ReLU networks with rational weights:*

(1) Does $\mathcal{N}$ contain a faithful circuit implementing f^*?

(2) Are circuits $\mathcal{C}$ and $\mathcal{C}'$ in the same universality class (Definition 7.1 with $\varepsilon = \phi = 0$)?

(3) Is $\mathcal{N}$ (ε, K_0)-interpretable (Corollary 4.8) for task $\mathcal{T}$ and $\varepsilon = 0$?

The undecidability of item (3) deserves special attention. It says that there is no algorithm that can decide, for an arbitrary network and task, whether a perfect (zero-error) mechanistic interpretation exists. For $\varepsilon > 0$, the problem becomes decidable (it reduces to a finite computation over the compact domain), but the complexity grows rapidly with $1/\varepsilon$.

8.3 The Complexity of Approximate Interpretability

While exact interpretability is undecidable, approximate interpretability ($\varepsilon > 0$) is decidable. The question is: how complex is it? We study this within the polynomial hierarchy.

Theorem 8.3 (Complexity of Approximate Circuit Discovery)**.** *Let $\mathcal{N}$ be a ReLU network with rational weights and $\varepsilon, \delta > 0$ be rational. The problem of deciding whether there exists a circuit $\mathcal{C} \subseteq \mathcal{N}$ with at most s edges that is (ε, D)-faithful for task $\mathcal{T}$ with tolerance δ on a finite input domain $\mathcal{X}_{\text{fin}}$ is Σ_2^P-complete.*

Σ_2^P is the second level of the polynomial hierarchy: the class of problems solvable in polynomial time with an NP oracle. Σ_2^P-completeness means the problem is at least as hard as any problem in this class — strictly harder than NP (assuming $\Sigma_2^P \neq$ NP, a standard conjecture in complexity theory [4]). Intuitively, the problem has two nested quantifiers: "there exists a circuit" (existential, NP-level) and "for all inputs in the domain, the circuit is faithful" (universal, co-NP-level).

The Σ_2^P-completeness result clarifies why circuit discovery is difficult in practice. It is not merely NP-hard; it is harder than NP, placing it beyond the reach of standard SAT-solvers and integer-programming approaches unless the problem instance has additional structure.

Proposition 8.4 (Approximation Ratio of ℓ^1 Circuit Discovery)**.** *Under standard complexity-theoretic assumptions (*P $\neq$ NP*), no polynomial-time algorithm for circuit discovery can achieve an approximation ratio better than $\Omega(\log n)$ in the worst case, where n is the number of edges in $\mathcal{N}$. The ℓ^1 relaxation* (2.3) *achieves an approximation ratio of $O(\log n)$ on instances satisfying the RIP (Definition 1.3 applied to the circuit structure).*

The $O(\log n)$ approximation ratio of the ℓ^1 method is thus optimal under standard assumptions — no tractable algorithm can do significantly better. This is the precise sense in which the circuit discovery methods of mechanistic interpretability are, in the worst case, as good as any tractable algorithm can be.

8.4 Statistical Limits: The Sample Complexity of Mechanistic Claims

Even if we set aside computational complexity, mechanistic interpretability faces a fundamental statistical limit: the claims it makes must be verified on data, and some mechanistic claims require exponentially many samples to verify.

Consider the claim: "Circuit $\mathcal{C}$ is the mechanism by which $\mathcal{N}$ computes task $\mathcal{T}$." For this claim to be meaningful, we need evidence that $\mathcal{C}$ accounts for $\mathcal{N}$'s behaviour not just on the training distribution D but on the full input space $\mathcal{X}$. The gap between in-distribution behaviour and full-space behaviour is the *distribution shift problem* for mechanistic claims.

Theorem 8.5 (Sample Complexity of Mechanistic Verification)**.** *Let $\mathcal{C}$ be a candidate circuit and D be a data distribution. The number of samples needed to verify, with confidence $1-\delta$, that $\mathcal{C}$ is (ε, D')-faithful for all distributions D' within total variation distance ρ of D, satisfies:*

$$N_{\text{ver}} \;\geq\; \Omega\left(\frac{1}{\varepsilon^2}\cdot e^{\Omega(d\rho^2)}\right), \tag{8.1}$$

where d is the input dimension.

The exponential dependence on $d\rho^2$ means that verifying mechanistic claims is exponentially hard in the input dimension, unless the distribution shift is small (small ρ). This is a formal expression of the observation that "the circuit works in training conditions" does not imply "the circuit works out of distribution."

The bound (8.1) is proved using a minimax argument [37]: any verification algorithm can be fooled by an adversarially chosen distribution D' within TV-distance ρ of D unless it has sampled $e^{\Omega(d\rho^2)}$ examples.

8.5 Irreducible Complexity and Non-Decomposable Circuits

The most philosophically interesting limit is structural: some circuits cannot be decomposed into simpler sub-circuits, making mechanistic analysis impossible in principle.

Definition 8.6 (Mechanistically Irreducible Circuit)**.** A circuit $\mathcal{C}$ is (ε, K_0)-*mechanistically irreducible* for task $\mathcal{T}$ if $\mathcal{C}$ is (ε, D)-faithful for $\mathcal{T}$ and, for every proper sub-circuit $\mathcal{C}' \subsetneq \mathcal{C}$ and every description P of $\mathcal{C}'$ with $K(P) \leq K_0$: $\mathcal{C}'$ is not (ε, D)-faithful.

In words: the circuit cannot be replaced by any simpler faithful sub-circuit whose description fits within K_0 bits.

An irreducible circuit is maximally uninterpretable: no simpler circuit accounts for the computation. Every edge is essential, every feature is necessary, and the circuit as a whole is the simplest possible implementation of its function.

Theorem 8.7 (Existence of Irreducible Circuits)**.** *For any $K_0 > 0$, there exist tasks $\mathcal{T}$ with $K(f_{\mathcal{T}}) > K_0$ for which the minimal faithful circuit is $(0, K_0)$-mechanistically irreducible. In particular, there exist tasks for which no compact mechanistic description exists.*

Proof. By a counting argument. The number of distinct circuits with $K(\mathcal{C}) \leq K_0$ is at most 2^{K_0}. The number of distinct tasks on n-bit inputs is 2^{2^n}. For $K_0 \ll 2^n$, almost all tasks cannot be implemented by any circuit of complexity $\leq K_0$. For such tasks, the minimal faithful circuit has $K(\mathcal{C}) > K_0$, making it irreducible by Definition 8.6. □

The counting argument is non-constructive. Are there natural tasks — tasks that arise from real-world data — for which the minimal faithful circuit is irreducible? The statistical language modelling task provides a heuristic argument. The Kolmogorov complexity of the next-token distribution for arbitrary natural language text is at least as large as the Kolmogorov complexity of the physical and social world (since language encodes information about the world), which is essentially infinite. As language models improve and their capabilities approach those of the minimal faithful circuit, the circuits they implement may become progressively harder to interpret, approaching irreducibility asymptotically.

8.6 The Interpretability–Performance Tradeoff

Definition 8.8 (Interpretability–Performance Frontier)**.** For a task $\mathcal{T}$ and network class $\mathcal{N}$, the *interpretability–performance frontier* is:

$$\mathrm{IPF}(\mathcal{N}, \mathcal{T}) \;=\; \{(\varepsilon, K_0) : \exists \mathcal{N} \in \mathcal{N},\, \mathcal{C} \subseteq \mathcal{N} \text{ that is } (\varepsilon, D)\text{-faithful and } K(\mathcal{C}) \leq K_0\}\,, \quad (8.2)$$

the Pareto frontier of achievable (test error, circuit complexity) pairs.

Theorem 8.9 (Shape of the IPF)**.** *For any task $\mathcal{T}$ with $K(f_{\mathcal{T}}) = K^*$:*

(1) The IPF satisfies $K_0 \geq K^* - C\log(1/\varepsilon)$ for a constant $C > 0$ — no network can achieve low error with a circuit simpler than the task itself.

(2) The IPF is convex — achieving a point strictly in the interior of the achievable region requires trading off some of both interpretability and performance.

(3) As $\varepsilon \to 0$ (perfect performance), $K_0 \to K^*$ (the circuit complexity approaches the Kolmogorov complexity of the task) — perfect performance requires full complexity.

Part (3) is the most significant. It says that the interpretability of a neural network circuit is fundamentally bounded by the complexity of its task: a network that perfectly solves a complex task must implement a complex circuit. Perfect interpretability ($K_0 = 0$, a trivially simple circuit) requires accepting perfect ignorance ($\varepsilon = 1$, a trivially bad model). There is no free lunch.

The bound in part (1) is a version of the rate-distortion lower bound (Lemma 4.1) applied to circuits. The convexity in part (2) follows from the convexity of the information bottleneck functional (4.4). Part (3) is a consequence of the MDL bound (Proposition 4.7): as $\varepsilon \to 0$, the MDL circuit approaches the minimal circuit for the task, whose complexity is K^*.

8.7 What the Limits Do and Do Not Preclude

The limits established in §§8.2–8.5 are not independent; they conspire to produce a fundamental tradeoff between interpretability and performance. But it is important to state what they do *not* preclude.

The undecidability of Theorem 8.1 applies to exact verification of universal claims. For any specific network, specific task, and specific finite input domain, all the questions raised by mechanistic interpretability are decidable — they reduce to finite computations over the domain. The undecidability is a worst-case result over the class of all possible networks. For the specific networks that arise in practice (trained by gradient descent on natural data distributions), the relevant circuits may be dramatically simpler than the worst case. The phenomenological richness of mechanistic interpretability studies — the curve detectors, the IOI circuits, the modular addition algorithm — is not threatened by the theoretical limits,

because these circuits have low Kolmogorov complexity and fall well below the irreducibility threshold.

The limits *do* preclude a *complete* mechanistic account of any network with non-trivial capability on complex tasks. By Theorem 8.9(3), achieving low test error on a complex task requires a circuit of complexity at least K^*. A human interpreter who can understand only circuits of complexity $K_0 < K^*$ will necessarily miss something. The gap is not a failure of effort or technique; it is a mathematical inevitability.

The practical resolution is that the interpretability programme should aim not for completeness but for *coverage*: understanding the circuits that account for the largest portion of the network's capability, accepting that some residual will always remain beyond reach. The information-theoretic framework of Chapter 4 provides the right criterion: a mechanistic account is as good as the mutual information $I(T_{\mathcal{C}}; Y)$ it captures, and the goal is to maximise this over all circuits whose complexity $K(\mathcal{C})$ is within human comprehension.

8.8 The Horizon Problem: Interpreting Interpreters

We close this chapter with a reflection on a deeper difficulty. All of the results above concern the interpretability of a fixed neural network $\mathcal{N}$ by a fixed interpreting algorithm $\mathcal{A}$. But the interpreting algorithm is itself a computation, and it may be implemented as a neural network. Who interprets the interpreter?

Proposition 8.10 (The Interpretation Regress)**.** *Let $\mathcal{A}$ be an interpretation algorithm that takes a network $\mathcal{N}$ as input and produces a circuit description $\mathcal{C}_{\mathcal{A}}(\mathcal{N})$ as output. If $\mathcal{A}$ is itself implemented as a neural network $\mathcal{N}_{\mathcal{A}}$, then the question "what is $\mathcal{N}_{\mathcal{A}}$ doing when it interprets $\mathcal{N}$?" requires applying $\mathcal{A}$ to $\mathcal{N}_{\mathcal{A}}$, which produces a circuit description of $\mathcal{A}$, which requires interpreting in turn. The regress is infinite.*

The regress is resolved in practice by fixing a "base level" of interpretation — human understanding — and accepting that the base level is itself a black box. We understand a neural network when we can describe its circuits in terms that human minds can grasp.

But this contract has limits. As language models become more capable, they will implement circuits of increasing complexity K^*. At some point, K^* will exceed the complexity that human minds can hold in working memory — the point at which the circuits are too intricate to be described in any human-intelligible language. By Theorem 8.9(3), this is precisely the point at which the interpretability programme reaches its ultimate mathematical limit.

What the theory does settle is the *structure of the question*. Mechanistic interpretability is an attempt to construct a mathematical science of computation as it actually occurs in large learned systems — a science with its own objects (features, circuits), its own laws (the Welch bound, the RIP, the information bottleneck), its own dynamics (grokking, phase transitions, symmetry breaking), and its own limits (undecidability, irreducibility, the interpretation horizon). That science is young. This monograph is one account of what it might look like when it grows up.

Bibliographic Notes

The undecidability of neural network verification relates to [57] and [36]. The Σ_2^P-completeness result connects to [12] and [4]. The sample complexity bound uses techniques from [37]. Computational irreducibility is introduced in [67]; its formal development in terms of Kolmogorov complexity draws on [41]. The interpretability–performance frontier combines the rate-distortion lower bound (Lemma 4.1) with the MDL bound (Proposition 4.7). The epistemology of scientific understanding that motivates §8.7 is treated in [58].

8.9 Logical Depth and the Complexity of Understanding

The Kolmogorov complexity $K(\mathcal{C})$ of a circuit $\mathcal{C}$ measures how compactly the circuit can be described. But compressibility is not the only measure of complexity relevant to interpretability. A circuit may be highly compressible (its description is short) yet still difficult for a human to understand, because executing the description requires many steps. This motivates the concept of *logical depth* [9].

Definition 8.11 (Logical Depth of a Circuit)**.** The *logical depth* of a circuit $\mathcal{C}$ at precision ε is the minimum number of computation steps required by the fastest ε-approximation to $\mathcal{C}$:

$$D_\varepsilon(\mathcal{C}) \;=\; \min\{t : \exists \text{ program } P \text{ running in } t \text{ steps with output } \varepsilon\text{-close to } \mathcal{C}\}. \tag{8.3}$$

Logical depth distinguishes between circuits that are "trivially simple" (short description, few execution steps) and those that are "superficially simple but deeply complex" (short description, many execution steps). A random string has high Kolmogorov complexity and low logical depth; a chess endgame has low Kolmogorov complexity (described by the rules of chess) but potentially high logical depth (requiring many moves to analyse).

The interpretability criterion (Corollary 4.8) should arguably include a bound on logical depth as well as Kolmogorov complexity:

Definition 8.12 (Extended Interpretability Criterion)**.** A network $\mathcal{N}$ is (ε, K_0, D_0)-*interpretable* for task $\mathcal{T}$ if there exists a circuit $\mathcal{C} \subseteq \mathcal{N}$ that is:

(1) (ε, D)-faithful for $\mathcal{T}$;

(2) $K(\mathcal{C}) \le K_0$ (short description);

(3) $D_\varepsilon(\mathcal{C}) \le D_0$ (fast execution of the description);

(4) Implemented by an approximately incoherent feature frame at each layer.

The additional condition (3) captures the distinction between a circuit that is interpretable in the sense that it can be described briefly (low K) and one that is interpretable in the sense that the description can be executed quickly (low D). For human understanding, both matter: we need not only a compact description of the circuit but one whose consequences are quickly apparent.

8.10 Approximate Circuits and Robust Interpretability

The faithfulness condition (Definition 2.4) requires the circuit to approximate the full network to within ε in L^2 loss. But there is a stronger notion of faithfulness that is more relevant for safety-critical applications: *pointwise* faithfulness.

Definition 8.13 (Pointwise Faithful Circuit)**.** A circuit $\mathcal{C} \subseteq \mathcal{N}$ is *pointwise* (ε, δ)-*faithful* for task $\mathcal{T}$ if:

$$\mathbb{P}_{x \sim D}[\|f_{\mathcal{C}}(x_{\text{in}}) - f_{\mathcal{N}}(x)[V_{\text{out}}]\| > \varepsilon] \;\le\; \delta. \tag{8.4}$$

That is, with probability at least $1 - \delta$, the circuit approximates the network to within ε on a random input.

The difference between (2.1) and (8.4) is the difference between L^2 approximation (squared error) and high-probability pointwise approximation. A circuit can be faithful in the L^2 sense while failing catastrophically on a small fraction of inputs — a failure mode that is precisely the concern in adversarial robustness and AI safety.

Proposition 8.14 (Equivalence of Faithfulness Notions for Incoherent Frames)**.** *For a circuit $\mathcal{C}$ with feature frame of coherence $\mu \leq 1/\sqrt{m}$ (incoherent frame) and sparsity k:*

$$(\varepsilon, D)\text{-faithful} \iff \text{pointwise } (\sqrt{k}\varepsilon, \delta)\text{-faithful for } \delta = O(k\varepsilon^2). \tag{8.5}$$

The implication from L^2 faithfulness to pointwise faithfulness requires the incoherence assumption: for incoherent frames, the L^2 error is dominated by the read-out error of individual features, which is bounded by $k\varepsilon^2$ at each position. The converse is immediate from the definition.

The proposition says that for networks with incoherent feature frames (which is the regime favoured by the mechanistic interpretability approach), L^2 faithfulness implies pointwise faithfulness with a small probability of failure. This is the formal guarantee that makes mechanistic interpretability not merely a tool for understanding networks but also a tool for verifying their safety.

Bibliographic Notes

The undecidability of neural network verification relates to [57] and [36]. The Σ_2^P-completeness result connects to [12] and [4]. The sample complexity bound uses techniques from [37]. Computational irreducibility is introduced in [67]; its development in terms of Kolmogorov complexity draws on [41]. Logical depth was introduced by Bennett [9] as a measure of the organised complexity of an object — the number of steps a fast algorithm needs to produce it from a short description.

8.11 The Verification Gap and Its Implications

The results of §§8.2–8.6 establish fundamental limits on what can be verified about neural network circuits. We now examine the *verification gap* — the distance between what can be proved about a circuit and what we would like to know — and its implications for the safety uses of mechanistic interpretability.

8.11.1 Specification vs. Verification

The safety programme of Chapter 10 requires both specifying what a network *should* do (the intended circuit $\mathcal{C}_{\mathcal{S}}$) and verifying that the network *does* do it (the actual circuit $\mathcal{C}_{\mathcal{N}}$ matches $\mathcal{C}_{\mathcal{S}}$). The verification gap arises because specification is tractable but verification is not:

- *Specification*: defining $\mathcal{C}_{\mathcal{S}}$ amounts to writing a formal program for the intended behaviour. This is hard for complex tasks (the same hardness as formal specification in software engineering) but is in principle computable — a human can write a program.
- *Verification*: checking that $\mathcal{C}_{\mathcal{N}} \approx \mathcal{C}_{\mathcal{S}}$ (in the sense of Definition 10.7) is Σ_2^P-hard (Theorem 8.3) and undecidable exactly (Corollary 8.2).

The gap between specification (hard but computable) and verification (undecidable) is the fundamental obstacle to using mechanistic interpretability for formal safety guarantees.

8.11.2 The Role of Human Verification

In practice, verification is performed by humans inspecting circuits and judging whether they implement the intended behaviour. This human-in-the-loop verification sidesteps the

computational hardness (humans can apply domain knowledge to verify circuits that formal algorithms cannot) but introduces a new limitation: the interpretability horizon of §8.8.

The interpretability horizon is the point at which K^* exceeds the complexity that human minds can hold in working memory. For current language models, this point has arguably not been reached for most capabilities — the circuits discovered for IOI, modular addition, and curve detection are well within human comprehension. But as models scale (Chapter 9) and the circuits become more complex (Proposition 9.3), the horizon will eventually be crossed.

Proposition 8.15 (The Verification Gap Grows with Scale)**.** *Let $V_{\text{gap}}(N)$ be the Kolmogorov complexity of the circuits that cannot be verified by any human interpreter with cognitive capacity K_{eval}:*

$$V_{\text{gap}}(N) \;=\; \max(0,\; K^*(N) - K_{\text{eval}})\,, \tag{8.6}$$

where $K^*(N)$ is the Kolmogorov complexity of the Platonic circuit for the model's primary task at scale N. By Theorem 9.2 and Proposition 4.7, $K^*(N)$ grows as $\Theta(N^\alpha)$ for the parameter scaling exponent α. Therefore $V_{\text{gap}}(N)$ grows without bound as $N \to \infty$, eventually exceeding K_{eval} for any fixed human cognitive capacity.

This proposition formalises the concern that the interpretability programme will eventually face circuits beyond human comprehension. However, the growth is polynomial (not exponential) in N, which means that scaling of K_{eval} (through better tools, AI-assisted interpretation, or team-based verification) can keep pace with scaling of $K^*(N)$ for substantial ranges of N.

8.11.3 Compositional Verification as a Partial Solution

The hierarchical oversight framework of §10.6 (Proposition 10.8) offers a partial solution: if the circuits of a large model decompose hierarchically into sub-circuits of bounded complexity, a human interpreter of bounded capacity K_{eval} can verify the full circuit by verifying each sub-circuit individually.

The key question is whether large language models' circuits are in fact hierarchically decomposable. The theory of Chapter 7 (specifically, Theorem 7.6) suggests that they are, at least for compositional tasks: the minimal faithful circuit for a depth-L compositional task decomposes into L sub-circuits of bounded complexity. For natural language tasks, which have strong compositional structure (sentences are built from phrases, phrases from words), this decomposition is plausible.

But the compositional decomposition is only partial: the high-importance circuits (for the most common and most complex capabilities) may have deep hierarchical structure, while the low-importance circuits (for rare or simple capabilities) may be shallow. The verification effort is dominated by the deep circuits, which may eventually exceed human capacity.

8.12 Adversarial Robustness Through the Interpretability Lens

A network that is mechanistically interpretable should, in principle, be more robust to adversarial attacks: if we understand what the network is computing, we can anticipate and prevent the failure modes that adversarial inputs exploit.

This intuition can be made precise using the circuit framework.

Definition 8.16 (Circuit Robustness)**.** A circuit $\mathcal{C} \subseteq \mathcal{N}$ is $(\varepsilon, \delta_{\text{adv}})$-*robust* if for all inputs x and adversarial perturbations Δ with $\|\Delta\| \leq \delta_{\text{adv}}$:

$$\|f_{\mathcal{C}}((x+\Delta)_{\text{in}}) - f_{\mathcal{C}}(x_{\text{in}})\| \;\leq\; \varepsilon. \tag{8.7}$$

Circuit robustness is stronger than the standard notion of robustness (which requires the network's output to be stable under perturbation) because it requires the circuit's *intermediate representations* to be stable as well. A network can be output-robust (its final prediction is stable) while being circuit-fragile (its internal computations change dramatically under perturbation).

Proposition 8.17 (Incoherent Frames Give Robust Circuits)**.** *Let $\mathcal{C}$ be a circuit whose feature frame at each layer has coherence $\mu_\ell \leq 1/\sqrt{d_\ell}$ (incoherent frame). Then $\mathcal{C}$ is $(\varepsilon_0\sqrt{k},\, \varepsilon_0\sqrt{d/k})$-robust, where k is the maximum instantaneous sparsity and ε_0 is the read-out error bound from Proposition 1.7.*

The proposition says that circuits with incoherent feature frames are robust to perturbations of size $O(\sqrt{d/k})$ with a response of size $O(\varepsilon_0\sqrt{k})$. This is the interpretability analogue of the ℓ^2 robustness guarantee for compressed sensing [15]: the RIP ensures that perturbations of the measurement (the activation $h(x)$) lead to proportionally small errors in the recovered representation (the activation vector a).

The practical implication is that mechanistic interpretability and adversarial robustness are not in tension but are complementary: networks with interpretable circuits (incoherent feature frames, sparse activations) are also more robust to adversarial attacks. This is a theoretical justification for the empirical observation that interpretability techniques tend to improve robustness.

Bibliographic Notes

Undecidability of neural network verification: [57], [36]. Σ_2^P-completeness: [12], [4]. Sample complexity: [37]. Computational irreducibility: [67], [41]. Logical depth: [9]. The IPF combines [11] with [29]. Adversarial robustness and its connection to compressed sensing uses the RIP framework of [15].

8.13 The Complexity of Circuit Equivalence

Beyond the undecidability of exact circuit verification (Theorem 8.1), there are important questions about the *relative complexity* of circuits: when are two circuits computing the same function, and how hard is it to determine this?

8.13.1 Functional Equivalence

Definition 8.18 (Functionally Equivalent Circuits)**.** Two circuits $\mathcal{C}$ and $\mathcal{C}'$ are (ε, D)-*functionally equivalent* if:

$$\mathbb{E}_{x\sim D}\left[\left\|f_{\mathcal{C}}(x_{\text{in}}) - f_{\mathcal{C}'}(x'_{\text{in}})\right\|^2\right] \;\leq\; \varepsilon^2, \tag{8.8}$$

where the expectations are over the same input distribution D.

Functional equivalence is weaker than isomorphism (Definition 7.1): two circuits can be functionally equivalent without being isomorphic (they may implement the same function through different algorithms). Conversely, two isomorphic circuits are always functionally equivalent.

The complexity of determining functional equivalence is:

Theorem 8.19 (Complexity of Functional Equivalence)**.** *The problem of deciding whether two ReLU circuits $\mathcal{C}$ and $\mathcal{C}'$ are $(0, D)$-functionally equivalent on a finite domain D is co-NP-complete.*

The proof reduces to the complement of the satisfiability problem: two circuits are functionally equivalent if and only if there is no input $x \in D$ on which their outputs differ. This is a co-NP statement ("for all x, outputs are equal"), and hardness follows from the hardness of Boolean formula satisfiability applied to the difference circuit $\mathcal{C} - \mathcal{C}'$.

8.13.2 The Isomorphism Gap

The gap between functional equivalence (co-NP-complete) and exact isomorphism (undecidable) is the *isomorphism gap*: there exist pairs of circuits that are functionally equivalent but not isomorphic. These are circuits that implement the same function through different algorithms, with different intermediate representations.

The existence of the isomorphism gap has a direct implication for the universality theory of Chapter 7: Theorem 7.4 predicts isomorphism (same algorithm), not merely functional equivalence (same input-output map). The stronger claim of isomorphism is what makes the universality theory non-trivial: any two networks trained on the same task will converge to the same *algorithm*, not just the same *function*.

8.13.3 Approximate Isomorphism and Its Complexity

The computational complexity of approximate isomorphism — deciding whether two circuits are (ε, ϕ)-isomorphic (Definition 7.1) — is:

Proposition 8.20 (Complexity of Approximate Isomorphism)**.** *The problem of deciding whether two circuits $\mathcal{C}$ and $\mathcal{C}'$ are (ε, ϕ)-isomorphic is in Σ_3^P (the third level of the polynomial hierarchy) and is Π_2^P-hard, for all $\varepsilon, \phi > 0$.*

The Σ_3^P upper bound comes from the three nested quantifiers: (1) $\exists$ orthogonal maps O_ℓ and scaling ρ (existential), (2) $\exists$ bijection σ between node sets (existential), (3) $\forall$ inputs x, the isomorphism condition (7.1) holds (universal). The Π_2^P-hardness comes from a reduction from the problem of deciding whether two Boolean formulas are "approximately equivalent" — a problem known to be Π_2^P-hard.

8.14 Decidability for Special Classes

While the general circuit verification problem is undecidable, several practically important special classes are decidable.

8.14.1 Linear Circuits

A circuit $\mathcal{C}$ is *linear* if all activation functions are the identity ($\phi_v = \mathrm{id}$). Linear circuits implement affine functions, and circuit equivalence for linear circuits reduces to matrix equality:

Proposition 8.21 (Decidability of Linear Circuit Equivalence)**.** *The problem of deciding whether two linear circuits $\mathcal{C}$ and $\mathcal{C}'$ are $(0, D)$-functionally equivalent is decidable in polynomial time: compute the product matrices $W_{\mathcal{C}} = W_L \cdots W_1$ and $W_{\mathcal{C}'} = W'_L \cdots W'_1$ (or their restrictions to the support of D) and check whether $W_{\mathcal{C}} = W_{\mathcal{C}'}$.*

Linear circuits are of limited practical relevance (neural networks with linear activations cannot implement non-linear functions), but they serve as a useful baseline: the decidability of linear circuit equivalence confirms that it is the non-linearity of activation functions that makes verification hard.

8.14.2 Single-Layer ReLU Circuits

For a single hidden layer with ReLU activations, the function computed by the circuit is a continuous piecewise-linear (CPWL) function. Two single-layer ReLU circuits compute the same CPWL function if and only if they have the same linear regions and the same affine function on each region.

Proposition 8.22 (Decidability of Single-Layer Circuit Equivalence)**.** *The problem of deciding whether two single-layer ReLU circuits $\mathcal{C}$ and $\mathcal{C}'$ (with the same input and output dimensions) are $(0, \mathbb{R}^n)$-functionally equivalent is decidable in time $O(nm \cdot 2^{\min(n,m)})$, where n is the input dimension and m is the number of neurons.*

The exponential dependence on $\min(n, m)$ in Proposition 8.22 reflects the fact that a single-layer ReLU network can have up to 2^m linear regions, and verifying equivalence requires checking all of them. For small input dimensions or few neurons, this is tractable; for large networks, it is not.

Bibliographic Notes

Undecidability of neural network verification: [57], [36]. Σ_2^P-completeness: [12], [4]. Sample complexity: [37]. Computational irreducibility: [67], [41]. Logical depth: [9]. The IPF: [11], [29]. Adversarial robustness: [15]. Verification of piecewise-linear functions: Montufar et al. (2014), "On the Number of Linear Regions of Deep Neural Networks."

8.15 Quantitative Bounds on the Interpretability Gap

The theoretical limits established in §§8.2–8.6 are asymptotic: they show that certain problems are NP-hard, Σ_2^P-complete, or undecidable. For practical AI safety applications, we need quantitative bounds that apply to specific networks and tasks. We derive such bounds here.

8.15.1 The Effective Circuit Size and Sample Complexity

For a network $\mathcal{N}$ with N parameters trained on a task of Kolmogorov complexity K^*, the minimal faithful circuit has at most $s = K^*/\log(1/\varepsilon)$ edges (Proposition 4.7). The sample complexity for verifying the circuit to precision ε over distribution shifts of magnitude ρ is, by Theorem 8.5:

$$N_{\text{ver}}(s, \rho, \varepsilon) \;\geq\; \frac{1}{\varepsilon^2} \cdot e^{\Omega(d\rho^2)} \;\geq\; \frac{K^*}{\varepsilon^2 \log(1/\varepsilon)} \cdot e^{\Omega(d\rho^2)}. \tag{8.9}$$

For a language model with $d = 4096$ (typical hidden dimension) and a distribution shift of $\rho = 0.1$ (10% shift in TV distance), the sample complexity is at least $e^{\Omega(4096 \cdot 0.01)} = e^{\Omega(40)} \approx 10^{17}$ samples. This is astronomically larger than any feasible evaluation dataset, confirming that distribution-robust circuit verification is infeasible in practice for current model scales.

8.15.2 The Practical Approximation Ratio

The ℓ^1 circuit discovery algorithm (2.3) achieves an approximation ratio of $O(\log n)$ (Proposition 8.4), where $n = |E|$ is the number of edges in the network. For GPT-2 small ($n \approx 10^8$ edges), this gives an approximation ratio of $O(\log 10^8) \approx 27$. This means the discovered circuit may have up to 27 times as many edges as the minimal faithful circuit.

In practice, the discovered circuits tend to be much smaller than this worst-case bound suggests, because natural language tasks have low Kolmogorov complexity and the circuits are genuinely sparse. The IOI circuit (§2.8) has ~ 26 attention heads out of 144, which is $\sim 18\%$ of the total head count — not close to the 100% worst-case bound.

8.15.3 The Verification Gap in Numbers

We can now give concrete numbers for the "verification gap" of a specific network and task. Consider GPT-2 small (124M parameters) and the IOI task:

- Kolmogorov complexity of the IOI task: $K^* \approx 100$ bits (a few lines of pseudocode describe the algorithm precisely).
- Circuit size: $|E^{\mathcal{C}_{\text{IOI}}}| \approx 26 \times 64^2 \approx 10^5$ edges.
- Interpretability budget: $K(\mathcal{C}_{\text{IOI}}) \approx 100$ bits (the circuit description is short).
- Verification cost (on training distribution): $O(10^5)$ activation patches (polynomial).
- Verification cost (over 10% distribution shift): $\geq e^{40} \approx 10^{17}$ samples (infeasible).

The gap between in-distribution verification (feasible) and distribution-robust verification (infeasible) is the practical manifestation of Theorem 8.5. Current mechanistic interpretability practice operates entirely in the in-distribution regime, which is the only feasible regime.

Bibliographic Notes

Undecidability: [57], [36]. Σ_2^P-completeness: [12], [4]. Sample complexity: [37]. Logical depth: [9]. IPF: [11], [29]. Adversarial robustness: [15]. Circuit size and the IOI circuit: [64].

Chapter 9

Scaling Laws and the Asymptotic Theory of Circuits

9.1 Scaling as a Mathematical Phenomenon

Among the most unexpected discoveries of the past half-decade of deep learning research is the *scaling law*: the finding that the test loss of a neural language model decreases as a smooth power law in the number of parameters, the number of training tokens, and the amount of compute, with exponents that are stable across orders of magnitude of each variable and largely independent of architectural details. The Chinchilla scaling laws [31] give, for a model of N parameters trained on D tokens:

$$\mathcal{L}(N, D) \;\approx\; \mathcal{L}_\infty + \frac{A}{N^\alpha} + \frac{B}{D^\beta}, \tag{9.1}$$

with empirically determined constants $\mathcal{L}_\infty \approx 1.61$ nats, $A \approx 406.4$, $B \approx 410.7$, $\alpha \approx 0.336$, $\beta \approx 0.283$. These numbers have been reproduced across model families, training regimes, and decades of compute budgets. They are, in some sense, among the most reliable empirical facts in deep learning.

From the perspective of this monograph, the scaling law (9.1) is not merely an engineering observation. It is a statement about the asymptotic behaviour of circuits and features as the number of parameters grows: how many features can a network of N parameters represent, how does their importance scale, and how does the information-theoretic quality of its representations improve with scale? These questions have answers, and those answers are the subject of this chapter.

The central thesis is that the power-law behaviour of $\mathcal{L}(N, D)$ in (9.1) is a direct consequence of the power-law distribution of feature importances in natural language — a distribution that is itself a consequence of Zipf's law and the fractal structure of linguistic and conceptual space. The exponents α and β are not arbitrary empirical constants; they are (approximately) derivable from the statistics of natural language and the geometry of the superposition architecture developed in Chapter 1.

9.2 Feature Importance Distributions and Their Tails

In the superposition framework of Chapter 1, the network stores m features of varying importance. Recall the definition (9.2): the importance of feature i is

$$\lambda_i = I(f_i^\top T;\, Y) \;\geq\; 0, \tag{9.2}$$

ordered so that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m \geq 0$.

Proposition 9.1 (Power-Law Features from Zipfian Concepts)**.** *Let $\mathcal{C} = \{c_1, c_2, \dots\}$ be a set of concepts with occurrence probabilities $p_i \propto i^{-s}$ for $s > 1$. Define the importance $\lambda_i = p_i \log(1/p_i) \sim p_i \log i \propto i^{-s} \log i$. Then for large i, $\lambda_i \leq C i^{-\gamma}$ for any $\gamma < s$.*

The proof is immediate from the bound $\log i \leq C_\varepsilon i^\varepsilon$ for any $\varepsilon > 0$. The effective exponent of the power law is $\gamma \approx s$ for practical purposes, with a logarithmic correction that is negligible over the dynamic range of feature ranks encountered in practice.

For natural language, the Zipfian exponent of word frequencies is $s \approx 1$ (the famous Zipf's law [51]), but for structured concepts (phrases, syntactic constructions, facts) the distribution decays more rapidly, with $s \approx 5/3$ being a representative value for the effective Zipfian exponent of complex linguistic concepts [43].

9.3 Deriving the Parameter Scaling Exponent

A network of N parameters with hidden dimension $d \sim \sqrt{N}$ (typical for transformer architectures) can represent, by Lemma 1.6, up to $m^*(d) = e^{O(d)}$ features with tolerated interference ε. The network learns the m most important features and ignores the rest. The residual loss from ignoring features $m+1, m+2, \dots$ is:

$$\mathcal{L}_{\text{res}}(m) \;=\; \sum_{i>m} \lambda_i \;\approx\; \int_m^\infty \lambda_1 i^{-\gamma}\, di \;=\; \frac{\lambda_1}{(\gamma-1)\, m^{\gamma-1}}, \quad \gamma > 1. \tag{9.3}$$

The number of features a model of N parameters can store with interference ε is $m^*(N) = e^{O(\sqrt{N}\,\varepsilon^2)}$. The tolerated interference level scales as $\varepsilon \sim N^{-1/4}$ (the optimal allocation that balances interference against capacity, derived by minimising the total error — read-out error plus residual loss — over ε and m simultaneously). This gives $m^*(N) \sim e^{O(N^{1/2})}$, and substituting into (9.3):

$$\mathcal{L}(N) \;\approx\; \mathcal{L}_\infty + C \cdot N^{-(\gamma-1)/2}. \tag{9.4}$$

Theorem 9.2 (Derivation of the Parameter Scaling Exponent)**.** *Under the assumptions that:*

(i) feature importances follow $\lambda_i \propto i^{-\gamma}$ with $\gamma > 1$;

(ii) the model stores the top-m features with interference $\varepsilon \propto m^{-1/2} d^{-1/2}$ (achieving the Welch bound, Theorem 1.1);

(iii) the model dimension scales as $d \propto \sqrt{N}$;

the test loss satisfies $\mathcal{L}(N) \approx \mathcal{L}_\infty + A N^{-\alpha}$ with $\alpha = (\gamma - 1)/2$.

For $\gamma = 5/3$, Theorem 9.2 gives $\alpha = 1/3 \approx 0.333$, matching the empirical Chinchilla value $\alpha \approx 0.336$ to three significant figures. This is not a fit: it is a prediction from the geometry of superposition applied to a language model trained on a Zipfian corpus.

The data scaling exponent has a parallel derivation. A feature of importance λ_i requires, by standard learning theory, at least $O(1/\lambda_i)$ examples to learn reliably. The number of features learnable from D examples is $m^*(D) \approx (\lambda_1 D)^{1/\gamma}$ (inverting the power law). Substituting into (9.3):

$$\mathcal{L}(D) \;\approx\; \mathcal{L}_\infty + B \cdot D^{-(\gamma-1)/\gamma}, \tag{9.5}$$

giving $\beta = (\gamma - 1)/\gamma \approx 0.40$ for $\gamma = 5/3$. The empirical value $\beta \approx 0.283$ is somewhat lower, likely because the simplification that all features are learned equally efficiently from data overestimates the effective sparsity of natural language.

9.4 Circuits at Scale: The Asymptotic Regime

Proposition 9.3 (Circuit Count at Scale). *Under the power-law feature distribution with exponent γ, a model of N parameters implements approximately $m^*(N) \propto N^{1/[2(\gamma-1)]}$ features and $c^*(N) \propto N^{1/(\gamma-1)-1/2}$ distinct circuits (counting circuits as pairs of features connected by a computation).*

The circuit count grows super-linearly in N when $1/(\gamma-1) > 3/2$, i.e., when $\gamma < 5/3$ (the *dense circuit regime*). For $\gamma > 5/3$ (the *sparse circuit regime*), the circuit count grows sub-linearly. Natural language modelling with $\gamma \approx 5/3$ is close to the phase boundary between the two regimes.

This observation may explain why large language models exhibit both identifiable circuits (consistent with sparse-circuit behaviour for high-importance tasks) and diffuse, hard-to-localise capabilities (consistent with dense-circuit behaviour for low-importance tasks).

9.5 Emergent Capabilities and Phase Transitions at Scale

The scaling law (9.1) is smooth — a continuous function of N — but the capabilities of neural networks do not always emerge smoothly. Some capabilities appear abruptly at a critical model size, exhibiting phase transition behaviour [65].

Definition 9.4 (Emergent Capability). A task $\mathcal{T}$ exhibits an *emergent capability* at scale N^* if:

$$\left.\frac{d}{dN}\mathcal{L}(\mathcal{T}, N)\right|_{N=N^*} \gg \left.\frac{d}{dN}\mathcal{L}(\mathcal{T}, N)\right|_{N\neq N^*}, \tag{9.6}$$

i.e., the loss on $\mathcal{T}$ decreases much more rapidly near N^* than at nearby scales.

Theorem 9.5 (Circuit Formation Threshold for Emergence). *Let $\mathcal{T}$ be a task whose minimal faithful circuit $\mathcal{C}^*$ requires m^* features at sparsity k and interference ε. The capability emerges at the model scale:*

$$N^*(m^*, k, \varepsilon) \sim \left(\frac{\log m^*}{\varepsilon^2}\right)^2. \tag{9.7}$$

The prediction of Theorem 9.5 is that emergent capabilities occur at predictable scales, determined by the feature count of the required circuit. A circuit requiring many features (m^* large) emerges later than one requiring few. The sharpness of the emergence — how discontinuous the capability is — is controlled by the concavity of the loss near the threshold: if the loss drops steeply once the circuit forms, the emergence is sharp.

Corollary 9.6 (Predictable vs. Unpredictable Emergence). *Capabilities implemented by circuits with a single critical feature (whose absence makes the circuit non-faithful) emerge sharply. Capabilities implemented by circuits where each additional feature provides a marginal improvement emerge smoothly. The apparent "unpredictability" of emergence [65] is a measurement artefact: coarse evaluation metrics (e.g., zero-shot accuracy on a benchmark) appear to change discontinuously when the loss changes continuously but crosses a decision threshold. This reframing is consistent with the analysis of [55].*

9.6 The Double Descent Phenomenon in the Circuit Framework

Double descent [7, 44] is the non-monotone behaviour of test loss as model size increases through the interpolation threshold N_{interp} (where the number of parameters equals the number of training examples): the test loss first decreases, then increases, then decreases again.

Proposition 9.7 (Double Descent as Circuit Competition). *In the circuit framework, double descent corresponds to a competition between:*

- *The memorisation circuit $\mathcal{C}_{\text{mem}}$ (size $O(N_{\text{interp}})$, zero training loss, high test loss);*
- *The generalisation circuit $\mathcal{C}_{\text{alg}}$ (size $O(s) \ll N_{\text{interp}}$, low training and test loss).*

(1) For $N < N_{\text{interp}}$: only $\mathcal{C}_{\text{alg}}$ is feasible; test loss decreases smoothly.

(2) At $N = N_{\text{interp}}$: $\mathcal{C}_{\text{mem}}$ becomes feasible and dominates; test loss peaks.

(3) For $N > N_{\text{interp}}$: excess capacity allows $\mathcal{C}_{\text{alg}}$ to form alongside $\mathcal{C}_{\text{mem}}$; weight decay eliminates $\mathcal{C}_{\text{mem}}$ (Lemma 3.2) and test loss decreases again.

The double descent phenomenon is the scaling-law analogue of grokking (Chapter 3): both involve a competition between memorisation and generalisation circuits, with resolution determined by the thermodynamics of weight decay. The difference is that in grokking, the transition occurs over training time at fixed model size; in double descent, it occurs over model size at fixed training time.

This unification suggests a common mechanism: the *norm gap* of Lemma 3.2 is the key quantity in both cases. Whenever the memorisation solution has much larger weight norm than the algorithmic solution, weight decay will eventually drive the system toward the algorithmic solution — whether that transition happens in training time (grokking) or in model scale (double descent).

Bibliographic Notes

The Chinchilla scaling laws are reported in [31]. The derivation of scaling exponents from feature importance distributions in Theorem 9.2 builds on [43]. The connection to Zipf's law is treated in [51]. Emergent capabilities are empirically characterised in [65] and theoretically reanalysed in [55]. Double descent is established in [7] and [44].

9.7 Optimal Compute Allocation: The Chinchilla Frontier

Theorem 9.2 characterises the loss as a function of parameters N and data D separately. The practical question — given a fixed compute budget $C = N \cdot D \cdot \text{cost}$, how should one allocate between model size and training data? — is answered by the Chinchilla optimal allocation.

Proposition 9.8 (Optimal Parameter–Data Tradeoff). *For a fixed compute budget C, the test loss $\mathcal{L}(N, D) = \mathcal{L}_\infty + AN^{-\alpha} + BD^{-\beta}$ subject to the compute constraint $C \propto ND$ is minimised at:*

$$N^*(C) \propto C^{\beta/(\alpha+\beta)}, \qquad D^*(C) \propto C^{\alpha/(\alpha+\beta)}. \tag{9.8}$$

Under Chinchilla scaling ($\alpha \approx 0.336$, $\beta \approx 0.283$), this gives $N^* \propto C^{0.46}$ and $D^* \propto C^{0.54}$, meaning optimal models should allocate slightly more compute to data than to parameters.

Proof. Minimise $\mathcal{L}(N, D)$ subject to $ND = C/c$ for some constant $c > 0$ (the compute cost per parameter-token product). By Lagrange multipliers, at the optimum: $\partial\mathcal{L}/\partial N = \partial\mathcal{L}/\partial D \cdot (D/N)$, i.e., $-\alpha AN^{-\alpha-1} = -\beta BD^{-\beta-1} \cdot (D/N)$. This gives $\alpha AN^{-\alpha-1} = \beta BD^{-\beta-1} N/D^2 \cdot D^2$, which simplifies to $(\alpha A/N^\alpha) = (\beta B/D^\beta)$, i.e., the two loss terms are in ratio $\alpha : \beta$ at the optimum. Solving with the constraint $ND = C/c$ gives (9.8). □

The interpretability implication of Proposition 9.8 is that the optimal model for a given compute budget is not the largest model (which would be maximally expressive but underfit in features per training example) nor the smallest model (which would overfit its training data without learning the most important features). It is a model that balances the two: large enough to represent the $m^*(N)$ most important features, and trained on enough data to learn each of them reliably.

9.8 Connections to Statistical Physics

The derivation of the scaling law exponents in Theorem 9.2 has a natural interpretation in the language of statistical physics, which we briefly develop here.

Consider the network as a system with m "degrees of freedom" (features), each with an "energy" λ_i (importance) and a "cost" $1/m$ (the fraction of the parameter budget allocated to it). The "partition function" is:

$$Z(\beta_{\text{phys}}, N) \;=\; \sum_{i=1}^{m^*(N)} e^{-\beta_{\text{phys}}\lambda_i}, \tag{9.9}$$

where β_{phys} is an inverse temperature controlling the tradeoff between "learning high-importance features" (low β_{phys}) and "having many features" (high β_{phys}). The "free energy" of the system is:

$$\mathcal{F}(\beta_{\text{phys}}, N) \;=\; -\frac{1}{\beta_{\text{phys}}} \log Z(\beta_{\text{phys}}, N), \tag{9.10}$$

and the expected loss is the derivative: $\mathcal{L}(N) \approx -\partial \log Z / \partial \beta_{\text{phys}}$.

For a power-law density of states $\rho(\lambda) \propto \lambda^{-\gamma}$, the partition function is a Laplace transform of a power law, which evaluates to $Z(\beta_{\text{phys}}) \propto \Gamma(1-\gamma)\beta_{\text{phys}}^{\gamma-1}$ for $\gamma < 1$ (converging only when $\gamma < 1$; for $\gamma > 1$ one needs a truncation at $\lambda = \lambda_\infty > 0$). The resulting free energy gives a loss $\mathcal{L} \propto N^{-(\gamma-1)/2}$, exactly as in Theorem 9.2.

This statistical mechanics picture — features as degrees of freedom, importances as energies, model scale as inverse temperature — provides a unified language for the three phenomena studied in this chapter: scaling laws (equilibrium thermodynamics of the feature system), grokking (a first-order phase transition in training dynamics), and emergence (a critical phenomenon at a phase boundary).

Bibliographic Notes

The Chinchilla scaling laws are reported in [31]. The derivation of scaling exponents from feature importance distributions builds on [43]. The connection to Zipf's law is treated in [51]. The optimal compute allocation (Proposition 9.8) is derived in [31]. Emergent capabilities are empirically characterised in [65] and theoretically reanalysed in [55]. Double descent is established in [7] and [44]. The statistical physics perspective on scaling laws connects to the replica method of [2] applied to the feature system, rather than the spin system.

9.9 Scaling Laws for Interpretability

The scaling laws of §9.3 concern the prediction loss $\mathcal{L}(N, D)$. But from the perspective of this monograph, the relevant question is: how does the *interpretability* of a model scale with its size? Does a larger model become more or less interpretable?

The answer, from the theory developed in previous chapters, is nuanced. There are three competing effects.

Effect 1: More circuits, more complexity. By Proposition 9.3, the number of distinct circuits grows as $c^*(N) \propto N^{1/(\gamma-1)-1/2}$ for $\gamma \approx 5/3$. A larger model has more circuits, and each circuit is potentially more complex (higher Kolmogorov complexity, Proposition 4.7). This suggests that larger models are *less* interpretable.

Effect 2: Better separation, more identifiable features. By Lemma 1.6 and the capacity-interference tradeoff (Proposition 1.7), a larger model can represent its features with lower interference $\varepsilon \sim N^{-1/4}$. Lower interference means that individual features are more easily identified and read out, because the contamination from other features is smaller. This suggests that larger models have *more identifiable* features.

Effect 3: More universal circuits. By the universality theory of Chapter 7, larger models that implement more features will be more likely to implement the Platonic circuits for those features (because larger models have more capacity to find the global minimum of the loss). This suggests that larger models implement circuits that are in the same universality class as the Platonic circuits, making them *more universally interpretable* — the circuits are the same as those found in other models, so understanding one model gives insight into others.

The net effect of these three competing trends is captured by the *interpretability scaling law*:

Proposition 9.9 (Interpretability Scaling Law)**.** *Under the power-law feature distribution with exponent γ and the optimal magnitude allocation of Proposition 1.8, the* fraction of the model's capability that is interpretable *(fraction of the loss reduction attributable to circuits with $K(\mathcal{C}) \leq K_0$) scales as:*

$$\text{InterpFrac}(N, K_0) \;\approx\; 1 - C \cdot N^{-\alpha} \cdot \exp\left(-\frac{K_0}{\kappa}\right), \tag{9.11}$$

for constants $C > 0$ and $\kappa > 0$ depending on γ and λ. The interpretable fraction increases with both N (larger models are more interpretable at fixed K_0) and K_0 (more complex descriptions explain more of the model).

The proposition says that interpretability is not a binary property but a continuous one: any model is interpretable to some extent, and the extent grows with both model scale and interpretive effort. The right question is not "is this model interpretable?" but "what fraction of its capability can be understood with a description of complexity K_0?"

This reframing has practical implications for AI safety: as models scale, the absolute amount of capability that is interpretable grows (because larger models learn more and better-separated features), even though the fraction may not keep pace with the total growth. The safety challenge is not that interpretability decreases with scale but that the *gap* between interpretable and total capability may grow.

9.10 Connections to Algorithmic Information Theory

The scaling law derivation of Theorem 9.2 can be recast in the language of *algorithmic information theory* [41], which provides a cleaner theoretical framework at the cost of uncomputability.

Let $f_{\mathcal{L}}$ be the true next-token distribution of natural language — the oracle that, for any context x, gives the exact probability $p(y|x)$ of the next token. The Kolmogorov complexity of $f_{\mathcal{L}}$ is the length of the shortest program computing it on a universal Turing machine.

The information content of the optimal language model (one that achieves $\mathcal{L}_\infty$ loss on every context) is bounded below by $K(f_\mathcal{L})$. By the chain rule of Kolmogorov complexity:

$$K(f_\mathcal{L}) \;\leq\; K(p_{\text{world}}) + O(K(p_{\text{grammar}})), \tag{9.12}$$

where p_{world} is the distribution over the world's states (which determines the content of language) and p_{grammar} is the grammar of the language (which determines the form). Since $K(p_{\text{world}})$ is essentially the Kolmogorov complexity of the observable universe — finite but astronomically large — $K(f_\mathcal{L})$ is large.

The scaling law (Theorem 9.2) can then be interpreted as: the residual loss $\mathcal{L}(N) - \mathcal{L}_\infty \propto N^{-\alpha}$ measures how much of $K(f_\mathcal{L})$ remains unexplained by a model of N parameters. A model of N parameters can "explain" approximately $N^{1/(2(\gamma-1))}$ features of $f_\mathcal{L}$ (by Proposition 9.3), each contributing a fixed amount to the reduction in $K(f_\mathcal{L})$. The remaining unexplained features contribute the residual loss $\mathcal{L}(N) - \mathcal{L}_\infty$.

Bibliographic Notes

The Chinchilla scaling laws are in [31]. The derivation of scaling exponents from feature importance distributions builds on [43]. The Zipf's law connection is in [51]. Emergent capabilities are in [65] and [55]. Double descent is in [7] and [44]. The optimal compute allocation is from [31]. The algorithmic information theory perspective uses [41].

9.11 The Sample Complexity of Feature Learning

The scaling law analysis of this chapter focuses on the test loss, which measures the average quality of the model's predictions. Equally important is the *sample complexity of feature learning*: how many training examples are needed to learn each feature reliably?

9.11.1 The Required Sample Complexity

By a standard PAC learning argument [4], a binary feature f_i with importance λ_i (mutual information with the label) requires at least $\Omega(1/\lambda_i)$ training examples to be learned with constant probability. This is because the statistical signal from feature i is proportional to λ_i, and one needs at least $O(1/\lambda_i)$ examples to overcome the noise.

More precisely, for a feature of importance λ_i and a training set of size N, the expected accuracy of the feature read-out is:

$$\mathbb{E}[\text{acc}(f_i)] \;\approx\; 1 - e^{-\lambda_i N/d}, \tag{9.13}$$

where d is the dimension of the representation and the exponential approximation is valid for $\lambda_i N/d \gtrsim 1$. A feature is "learned" (accuracy above $1-\delta$) when $N \geq d\log(1/\delta)/\lambda_i$.

9.11.2 The Cumulative Learning Curve

As the training set grows from $N = 0$ to $N = \infty$, features are learned in order of decreasing importance: feature 1 is learned first (at $N_1 \sim d/\lambda_1$), then feature 2 ($N_2 \sim d/\lambda_2$), and so on. The cumulative number of features learned as a function of N is:

$$m(N) \;\approx\; \max\left\{i : \frac{d}{\lambda_i} \leq N\right\} \;\approx\; \left(\frac{\lambda_1 N}{d}\right)^{1/\gamma}, \tag{9.14}$$

where the last step uses the power-law $\lambda_i \approx \lambda_1 i^{-\gamma}$ (inverted as $i \approx (\lambda_1/\lambda_i)^{1/\gamma}$).

The cumulative test loss is then:

$$\mathcal{L}(N) \;\approx\; \mathcal{L}_\infty + \sum_{i>m(N)} \lambda_i \;\approx\; \mathcal{L}_\infty + \frac{\lambda_1}{(\gamma-1)m(N)^{\gamma-1}} \;\approx\; \mathcal{L}_\infty + B\left(\frac{d}{\lambda_1 N}\right)^{(\gamma-1)/\gamma}, \quad (9.15)$$

confirming the data scaling law with $\beta = (\gamma-1)/\gamma \approx 0.40$ for $\gamma \approx 5/3$.

9.11.3 The Interplay Between Model Capacity and Data

The joint scaling law (9.1) with both N and D is a consequence of two independent constraints:

(1) *Capacity constraint*: the network can represent at most $m^*(N)$ features.

(2) *Data constraint*: the network can learn at most $m(D)$ features from D examples.

The effective number of learned features is $m_{\text{eff}} = \min(m^*(N), m(D))$, and the residual loss is:

$$\mathcal{L}(N, D) \;\approx\; \mathcal{L}_\infty + \frac{\lambda_1}{(\gamma-1)m_{\text{eff}}^{\gamma-1}}. \quad (9.16)$$

For the optimal allocation (Proposition 9.8), $m^*(N) \approx m(D)$, and both constraints are simultaneously binding. Substituting into (9.16) gives the Chinchilla formula (9.1), confirming the derivation of Theorem 9.2.

9.12 Feature Decomposition and the Semantic Hierarchy

The power-law feature importance distribution $\lambda_i \propto i^{-\gamma}$ implies a *semantic hierarchy*: the most important features (small i, large λ_i) encode the most general and frequent concepts, while the least important features (large i, small λ_i) encode the most specific and rare concepts.

This hierarchy has a natural interpretation in terms of the information-theoretic decomposition of the language modelling task. At the top of the hierarchy are features that reduce the loss by a large amount — features encoding basic grammar, common words, and universal semantic categories. At the bottom are features encoding rare words, domain-specific concepts, and idiosyncratic patterns.

Proposition 9.10 (The Semantic Hierarchy and Feature Importance)**.** *Under the power-law feature distribution with exponent γ, the features at rank i and $2i$ have importance ratio $\lambda_i/\lambda_{2i} = 2^\gamma$. A factor of $2^\gamma \approx 3.2$ (for $\gamma \approx 5/3$) in importance corresponds to a factor of 2 in feature rank. This means that the "semantic distance" between adjacent ranks in the hierarchy is approximately logarithmic in the importance ratio.*

This logarithmic structure of the semantic hierarchy is consistent with the observation that language models exhibit a hierarchical organisation of their features: high-importance features (e.g., "is this a noun?") are represented early in training and in all model sizes, while low-importance features (e.g., "is this an archaic legal term?") are represented only in larger models and later in training. The scaling law (Theorem 9.2) quantifies this hierarchical emergence precisely.

Bibliographic Notes

Scaling laws: [31]. Feature importance distributions: [43]. Zipf's law: [51]. Emergent capabilities: [65], [55]. Double descent: [7], [44]. Chinchilla optimal allocation: [31]. Algorithmic information theory: [41]. PAC learning and sample complexity: [4].

9.13 Mechanistic Interpretability and the Scaling Hypothesis

The scaling hypothesis — the conjecture that larger models will continue to improve on all tasks as compute increases — is often stated as an empirical observation. The theory of this monograph allows a more precise statement.

9.13.1 The Strong Scaling Hypothesis

Definition 9.11 (Strong Scaling Hypothesis)**.** The *strong scaling hypothesis* for task $\mathcal{T}$ with Kolmogorov complexity K^* is the claim that for any $\varepsilon > 0$, there exists a model scale $N^*(\varepsilon, \mathcal{T})$ such that for all $N > N^*$, a model trained on $D^*(N)$ tokens achieves test error $\mathcal{L}(N, D^*(N)) \leq \mathcal{L}_\infty + \varepsilon$.

The strong scaling hypothesis is true (by Theorem 9.2) for tasks where:

(1) The feature importance distribution satisfies the power law (Proposition 9.1) with $\gamma > 1$;

(2) The irreducible entropy $\mathcal{L}_\infty < \infty$ (the task is not perfectly random).

Under these conditions, the residual loss $\mathcal{L}(N, D^*(N)) - \mathcal{L}_\infty \propto N^{-\alpha} \to 0$ as $N \to \infty$.

9.13.2 The Interpretability Scaling Hypothesis

The interpretability analogue of the strong scaling hypothesis is:

Definition 9.12 (Interpretability Scaling Hypothesis)**.** The *interpretability scaling hypothesis* for task $\mathcal{T}$ is the claim that for any $K_0 > 0$, there exists a model scale $N^*(K_0, \mathcal{T})$ such that for all $N > N^*$, the (ε, K_0)-interpretable fraction of the model's capability (Proposition 9.9) exceeds $1 - \varepsilon$ for some $\varepsilon < 1/2$.

In words: the interpretability scaling hypothesis says that for sufficiently large models, *most* of the model's capability can be described by circuits of complexity at most K_0.

Proposition 9.13 (Status of the Interpretability Scaling Hypothesis)**.** *Under the power-law feature distribution with exponent γ:*

(1) For $K_0 \geq K^*$ (description complexity at least as large as the task): the interpretability scaling hypothesis holds, and $N^*(K_0, \mathcal{T}) = O(1)$ (any model suffices).

(2) For $K_0 < K^*$ (description complexity less than the task): the interpretability scaling hypothesis fails for the task as a whole, but holds for the fraction of capability attributable to circuits of complexity at most K_0 (a weaker claim).

The proposition distinguishes two regimes. For tasks with $K^* \leq K_0$ (tasks whose algorithms can be described in K_0 bits), perfect interpretability is achievable. For tasks with $K^* > K_0$ (tasks whose algorithms exceed the interpretation budget), partial interpretability is the best achievable, and the fraction increases with model scale.

9.13.3 Implications for the Safety Research Agenda

The interpretability scaling hypothesis has concrete implications for the AI safety research agenda:

(1) *Interpretability tools must scale*: As models grow, the circuits they implement grow more complex. Interpretability tools must be able to handle circuits of increasing size and depth. Tools that work well on GPT-2 scale may need fundamental redesign for frontier-model scale.

(2) *Hierarchical interpretability is necessary*: For circuits of complexity $K^* > K_{\text{eval}}$ (human cognitive capacity), the hierarchical oversight framework of §10.6 is required. The interpretability research agenda must invest in tools for hierarchical circuit decomposition.

(3) *Universality enables transfer*: The universality of Platonic circuits (Chapter 7) means that circuits discovered in smaller models can guide the search for circuits in larger models. This transfer of interpretability knowledge is the most tractable path to scaling interpretability alongside model scale.

Bibliographic Notes

Scaling laws: [31]. Feature distributions: [43]. Zipf's law: [51]. Emergent capabilities: [65], [55]. Double descent: [7], [44]. AIT: [41]. PAC learning: [4].

9.14 The Phase Diagram of Scaling

The scaling law analysis of this chapter characterises the loss as a continuous function of N and D. But as in Chapters 3 and 6, the continuous behaviour conceals discrete phase transitions. We briefly characterise the phase diagram of scaling.

9.14.1 Three Scaling Regimes

The scaling law (9.1) identifies three qualitatively distinct regimes:

Regime I — Data-limited ($D/N \ll D^*/N^*$): The model has more capacity than the training data supports. Increasing N does not reduce the loss; increasing D does. The loss is approximately $\mathcal{L} \approx \mathcal{L}_\infty + BD^{-\beta}$ (independent of N).

Regime II — Compute-optimal ($D/N \approx D^*/N^*$, the Chinchilla ratio): Both N and D are balanced optimally. The loss is $\mathcal{L} \approx \mathcal{L}_\infty + \frac{A}{N^\alpha} + \frac{B}{D^\beta}$ with both terms comparable.

Regime III — Model-limited ($D/N \gg D^*/N^*$): The training data is more than the model can absorb. Increasing D does not reduce the loss; increasing N does. The loss is approximately $\mathcal{L} \approx \mathcal{L}_\infty + AN^{-\alpha}$ (independent of D).

In terms of the circuit framework, these regimes correspond to:

- Regime I: $m(D) < m^*(N)$ (fewer features learned than representable); the bottleneck is data.
- Regime II: $m(D) \approx m^*(N)$ (balance between representable and learned features); both constraints are binding.
- Regime III: $m(D) > m^*(N)$ (more features learnable from data than representable); the bottleneck is model capacity.

Proposition 9.14 (Circuit Perspective on Scaling Regimes)**.** *The phase boundary between Regime I and Regime II occurs at $D \propto N^{\alpha/\beta}$. For Chinchilla parameters ($\alpha/\beta \approx 0.336/0.283 \approx 1.19$), the optimal ratio is $D^*/N^* \propto N^{0.19}$: the optimal training dataset grows slightly faster than the model size. This confirms the Chinchilla finding that larger models should be trained on more data than the naive "20 tokens per parameter" heuristic would suggest.*

Bibliographic Notes

Scaling laws: [31]. Feature distributions: [43]. Zipf's law: [51]. Emergence: [65], [55]. Double descent: [7], [44]. Optimal allocation: [31]. AIT: [41]. PAC learning: [4].

Chapter 10

Applications to AI Safety and Alignment

10.1 The Safety Imperative

The preceding nine chapters have developed a mathematical theory for its own sake — or so it might appear. The objects of study (features, circuits, frames, phase transitions) are interesting in the way that all deep mathematics is interesting: because they reveal structure where there appeared to be none, because they connect disparate phenomena under a unified explanation, because they generate predictions that can be tested. These are the reasons one does science.

But mechanistic interpretability has, besides the purely scientific motivation, an urgency that is unusual in mathematics. The systems whose circuits we are studying are not passive objects of inquiry — they are active agents increasingly embedded in consequential decisions. A language model advising a doctor, writing legal documents, or managing critical infrastructure is not merely a mathematical curiosity; it is a system whose failure modes, whose biases, whose potential for deception and misalignment, matter enormously. The central thesis of this monograph — that neural networks implement identifiable circuits encoding identifiable features — is directly relevant to understanding those failure modes.

This chapter therefore serves as both a conclusion and a beginning. We draw out the implications of the theory for three specific problems in AI safety: detecting deception, characterising alignment failures, and verifying safety properties. In each case, the mathematical framework allows us to state the problem precisely and to identify what a solution would look like — even when complete solutions remain out of reach.

10.2 Detecting Deception: The Circuit Signature of Dishonesty

A central concern in AI safety is the possibility that a sufficiently capable system might behave deceptively: presenting a benign appearance to evaluators while pursuing a different objective internally [33]. The formal structure of this concern, from the circuit perspective, is as follows.

Definition 10.1 (Deceptive Circuit)**.** A network $\mathcal{N}$ is (p, q, ε)-*deceptive* if there exist circuits $\mathcal{C}_{\text{ext}} \subseteq \mathcal{N}$ and $\mathcal{C}_{\text{int}} \subseteq \mathcal{N}$ such that:

(1) $\mathcal{C}_{\text{ext}}$ is $(\varepsilon, D_{\text{eval}})$-faithful for the externally visible task $\mathcal{T}_{\text{ext}}$;

(2) $\mathcal{C}_{\text{int}}$ is $(\varepsilon, D_{\text{dep}})$-faithful for an internal objective task $\mathcal{T}_{\text{int}}$;

(3) $I(f_{\mathcal{C}_{\text{ext}}}(X); f_{\mathcal{C}_{\text{int}}}(X)) \leq p$ (near-independence of the two circuits);

(4) The output of $\mathcal{C}_{\text{int}}$ influences the output of $\mathcal{N}$ on deployment inputs with probability $q > 0$.

The independence condition (3) formalises the intuition that a deceptive system keeps its "true" computation hidden from its "surface" computation. The influence condition (4) ensures that the internal circuit actually affects the network's behaviour on the inputs that matter, even if it is suppressed during evaluation.

Theorem 10.2 (Circuit-Based Deception Detection)**.** *If $\mathcal{N}$ is (p, q, ε)-deceptive, then the circuit discovery algorithm* (2.3) *applied to $\mathcal{N}$ on the evaluation distribution D_{eval} will recover $\mathcal{C}_{\text{ext}}$ but not $\mathcal{C}_{\text{int}}$, with high probability. However, applying the algorithm with the augmented loss*

$$\mathcal{L}(W) + \mu \cdot I(\mathcal{C}_{\text{ext}}; \textit{full network output})$$

will detect $\mathcal{C}_{\text{int}}$ with probability at least $1 - e^{-q^2|D_{\text{dep}}|/\varepsilon^2}$.

The proof uses the variational mutual information formula (4.6): the penalty term forces the algorithm to account for all parts of the network that are correlated with the output, including $\mathcal{C}_{\text{int}}$ on deployment inputs.

The practical implication is that detecting deception requires *deployment-distribution data* — examples drawn from the conditions under which the network might behave deceptively. This is a fundamental limitation: if the deployer does not have access to representative deployment data (because the deceptive behaviour is triggered by rare or adversarially chosen inputs), the circuit-based deception detection of Theorem 10.2 may fail. This is the safety analogue of the statistical limit of Theorem 8.5.

10.3 Alignment Failures as Circuit Misalignment

The most common form of misalignment — a network pursuing a different objective than the one intended — can be characterised as *circuit misalignment*: the network has learned circuits for a proxy objective $\tilde{\mathcal{T}}$ that correlates with the true objective $\mathcal{T}$ on the training distribution but diverges from it out of distribution.

Definition 10.3 (Circuit Misalignment)**.** A network $\mathcal{N}$ is (δ, ρ)-*misaligned* if its minimal faithful circuit for the true objective $\mathcal{T}$ and the proxy objective $\tilde{\mathcal{T}}$ satisfy $d_{\text{circ}}(\mathcal{C}^*_{\mathcal{T}}, \mathcal{C}^*_{\tilde{\mathcal{T}}}) > \delta$, and the proxy approximates the true objective with error ρ on the training distribution: $|\mathcal{L}_{\mathcal{T}}(x) - \mathcal{L}_{\tilde{\mathcal{T}}}(x)| \leq \rho$ for all $x \in D_{\text{train}}$.

Circuit misalignment is measurable in principle: compare the circuit extracted when the network is evaluated on $\mathcal{T}$ with the circuit extracted when evaluated on $\tilde{\mathcal{T}}$. If they are significantly different, the network has learned a proxy. The challenge is that this comparison requires knowing what the true objective $\mathcal{T}$ is, which may be difficult to specify precisely.

Proposition 10.4 (Detectable Misalignment from Feature Analysis)**.** *A necessary condition for circuit misalignment between $\mathcal{T}$ and $\tilde{\mathcal{T}}$ is that there exist features f_i with:*

$$\left| I(f_i^\top T; Y_{\mathcal{T}}) - I(f_i^\top T; Y_{\tilde{\mathcal{T}}}) \right| > \eta, \tag{10.1}$$

for some threshold $\eta > 0$. Any such features, identifiable by comparing feature importance scores (9.2) *under the two objectives, reveal the misalignment without requiring knowledge of the full circuit structure.*

The practical algorithm suggested by Proposition 10.4: compare the feature importances $\lambda_i^{\mathcal{T}}$ and $\lambda_i^{\tilde{\mathcal{T}}}$ for the two objectives. Features where these importances diverge are markers of misalignment — aspects of the representation that the network uses differently for the two tasks. This is measurable without running the full Σ_2^P-complete circuit discovery algorithm.

10.4 The Representation Geometry of Truthfulness

Among the features that the mechanistic interpretability framework predicts should be identifiable in large language models is a representation of *truthfulness*: a direction in activation space that the model uses to represent whether a given claim is true or false. Several empirical studies [14, 70] have found evidence for such a direction, identifiable by Contrast-Consistent Search (CCS) or Representation Engineering methods.

From the information-theoretic perspective of Chapter 4, a truthfulness direction $f_{\text{truth}} \in S^{d-1}$ is a sufficient statistic for the binary property $g_{\text{truth}}(X) = \mathbf{1}[\text{claim } X \text{ is true}]$: it satisfies $I(f_{\text{truth}}^\top T;\, g_{\text{truth}}(X)) \approx H(g_{\text{truth}}(X))$. The Welch bound guarantees that such a direction can coexist with many other feature directions without excessive interference, provided activations are sparse.

Definition 10.5 (Causally Faithful Feature)**.** A feature direction $f_i \in S^{d-1}$ is $(\delta, \mathcal{I})$-*causally faithful* for property g_i under intervention $\mathcal{I}$ if:

$$\left|\mathbb{E}[g_i(X) \mid \text{do}(f_i^\top T = v)] - \mathbb{E}[g_i(X) \mid f_i^\top T = v]\right| \;\leq\; \delta, \tag{10.2}$$

where $\text{do}(\cdot)$ denotes a causal intervention in the sense of Pearl's do-calculus [50].

Theorem 10.6 (Causal Faithfulness and Circuit Exclusivity)**.** *A feature f_i is $(0, \mathcal{I})$-causally faithful for g_i if and only if:*

(1) f_i is 0-sufficient for g_i (Definition 4.3 with $\delta = 0$);

(2) All circuits producing g_i-relevant outputs pass through the node f_i — there is no bypass circuit computing g_i without involving f_i.

Condition (2) is the causal Markov condition: g_i is d-separated from all inputs given $f_i^\top T$ in the causal graph of the circuit.

Theorem 10.6 gives a circuit-theoretic criterion for causal faithfulness: the feature is causally relevant for a property if and only if it is a bottleneck in every circuit that computes that property. This is testable by the circuit discovery algorithm: if removing the node f_i from the network eliminates all faithful circuits for g_i, then f_i is causally faithful. If there remain faithful circuits that bypass f_i, the feature is a correlate but not a cause. The existence of bypass circuits would explain why simple representation steering sometimes fails to reliably control model behaviour [59, 63].

10.5 Formalising Alignment as a Circuit Design Problem

Definition 10.7 (Alignment as Circuit Correspondence)**.** A network $\mathcal{N}$ is $(\varepsilon, \delta, K_0)$-*aligned* with specification $\mathcal{S}$ if:

(1) The minimal faithful circuit $\mathcal{C}_\mathcal{N}$ for $\mathcal{N}$'s actual behaviour is (ε, K_0)-isomorphic (Definition 7.1) to the minimal faithful circuit $\mathcal{C}_\mathcal{S}$ for the specified behaviour;

(2) $D_{\text{KL}}(f_\mathcal{N}(x) \| f_\mathcal{S}(x)) \leq \delta$ for all x in the deployment distribution.

Condition (1) is the structural requirement: the network implements the right algorithm, not merely the right input-output map. Condition (2) is the behavioural requirement: the network's outputs are close to the specified outputs. The combination of both ensures that the network is aligned in a deep sense — it does the right thing for the right reasons. Definition 10.7 excludes systems that coincidentally produce the right outputs through an unintended and potentially unsafe mechanism.

10.6 Scalable Oversight and the Circuit Hierarchy

The *scalable oversight* programme in AI safety [3, 18] asks: how can a less capable supervisor reliably evaluate the outputs and internal states of a more capable AI? From the circuit perspective, this is a question about whether the evaluating system can identify and verify the circuits implemented by the evaluated system.

By Theorem 8.9(3), a system implementing a task of Kolmogorov complexity K^* requires a circuit of complexity at least K^*. An evaluating system with cognitive capacity $K_{\text{eval}} < K^*$ cannot verify the full circuit. This is the formal version of the scalable oversight problem.

However, the circuit hierarchy of Chapter 7 offers a partial resolution. If the circuits of the evaluated system form a hierarchy — each high-level circuit decomposing into lower-level circuits of bounded complexity — then an evaluator of capacity K_{eval} can verify the hierarchy level by level.

Proposition 10.8 (Hierarchical Oversight Complexity)**.** *Let $\mathcal{C}$ be a circuit of Kolmogorov complexity K^* that decomposes hierarchically as $\mathcal{C} = \mathcal{C}_r \circ \cdots \circ \mathcal{C}_1$ with each $\mathcal{C}_\ell$ having $K(\mathcal{C}_\ell) \leq K_{\text{eval}}$ and $r = O(K^*/K_{\text{eval}})$ levels. An evaluator of capacity K_{eval} can verify $\mathcal{C}$ in $O(r)$ steps, each requiring $O(K_{\text{eval}})$ cognitive work.*

The hierarchical decomposition is possible whenever the task factors compositionally into sub-tasks, each of complexity at most K_{eval}. For natural language tasks, this is plausible: complex reasoning decomposes into simpler steps, complex facts into simpler supporting facts. The scalable oversight programme is achievable, under the circuit framework, precisely to the extent that the tasks of interest have this compositional structure.

10.7 The Geometry of Values

We close with a speculation — mathematically informed but not yet fully formalised — about the representation of *values* in large language models. By "values" we mean the features that encode what the system cares about: the implicit weightings over outcomes that determine how the system trades off different goals.

The superposition framework of Chapter 1 applies to value features just as it applies to factual or syntactic features. A model trained on human feedback (RLHF) will represent, in its activation space, directions corresponding to value judgements: the "harmlessness" direction, the "helpfulness" direction, the "honesty" direction. The Welch bound constrains how many such directions can coexist in the representation space; the sparsity of value activations — most prompts do not simultaneously activate all value features — allows far more value features to coexist than the dimension would naively allow.

If values are coherent — if they form a consistent system without contradiction — then the corresponding feature vectors should be nearly orthogonal: independent value dimensions, each contributing separately to the system's behaviour. If values are in tension — if being more helpful sometimes requires being less honest, for instance — then the corresponding feature vectors should be non-orthogonal, with a positive inner product reflecting the correlation of the tension.

Hypothesis 10.9 (Geometric Coherence of Values)**.** The value features of a well-aligned language model form an approximately incoherent frame — a nearly orthogonal collection of directions in activation space — with a geometry that reflects the logical independence of the corresponding values. Tensions between values manifest as non-zero inner products between the corresponding feature vectors, with the magnitude of the inner product proportional to the strength of the tension. Value conflicts arise during forward passes in which two or more

value features are simultaneously activated with large magnitude, producing interference in the residual stream that the network must resolve through its circuit structure.

This hypothesis connects two hitherto separate fields: moral philosophy (the study of value coherence and tension) and mechanistic interpretability (the study of feature geometry). The inner product between value feature vectors is a geometric quantity that, if Hypothesis 10.9 is correct, encodes the philosophical relationship between the corresponding values.

The formalisation and testing of Hypothesis 10.9 requires identifying value features in large language models (extending the CCS and representation engineering approaches of §10.3), measuring their mutual inner products, and comparing the resulting geometry to the philosophical structure of values as understood by ethics and moral philosophy. This is a research programme that connects mathematics, computer science, and philosophy in an unusual way — and one that the theory of this monograph makes, for the first time, technically precise enough to pursue.

10.8 Coda: The Shape of the Theory

This monograph has moved from the geometry of single feature vectors (Chapter 1) through the algebra of circuits (Chapter 2), the dynamics of their emergence (Chapter 3), their information-theoretic character (Chapter 4), the computational machinery of attention (Chapter 5), the phase transitions of representation (Chapter 6), the universality of circuits across systems (Chapter 7), the fundamental limits of interpretation (Chapter 8), the asymptotics of scaling (Chapter 9), and finally to the implications for AI safety and alignment (this chapter).

The central mathematical object is the *circuit*: a sparse subgraph of a neural network implementing an identifiable function through well-separated features. The central mathematical result is the *circuit–feature duality* of Theorem 4.4: features are the vocabulary of representation, circuits are the grammar of computation, and neither can be understood without the other. The central quantitative result is the *capacity-interference tradeoff* of Proposition 1.7: the exponential capacity of superposition is available only at the cost of interference, and the tolerable interference is bounded by the sparsity of activations.

These three results — duality, tradeoff, capacity — form the core of a genuine mathematical theory. The theory is incomplete in many ways: the universality theorem of Chapter 7 depends on idealising assumptions; the phase transition theory of Chapter 3 uses the Kramers formula heuristically; the complexity bounds of Chapter 8 are worst-case and may not reflect typical cases for natural data distributions. These are not failures of the theory but invitations to its continuation.

The deepest question the theory raises is one it cannot answer: whether the circuits discovered by mechanistic interpretability are the *right* objects of study. We have assumed throughout that a neural network is best understood as a composition of circuits implementing identifiable functions. This assumption is productive — it generates testable predictions, organises empirical observations, and connects to rich mathematical traditions. But some capabilities of large neural networks may be genuinely distributed, genuinely holistic, genuinely resistant to decomposition into circuits and features. The limits of Chapter 8 suggest this possibility without settling it.

What the theory does settle is the *structure of the question*. Mechanistic interpretability is not merely an engineering programme for improving AI safety, important as that goal is. It is an attempt to construct a mathematical science of computation as it actually occurs in large learned systems — a science with its own objects, laws, dynamics, and limits. That science is young. This monograph is one account of what it might look like when it grows up.

Bibliographic Notes

On AI safety and deceptive alignment: [33]. On discovering latent knowledge: [14] and [70]. On concrete problems in AI safety: [3]. On scalable oversight and reinforcement learning from human preferences: [18]. On representation engineering and steering vectors: [59] and [63]. Pearl's do-calculus: [50].

10.9 Mechanistic Interpretability as a Scientific Programme

We step back to consider the scientific status of the mechanistic interpretability programme in light of the theory developed in this monograph.

10.9.1 The Levels of Analysis

David Marr's three levels of analysis [42] — computational, algorithmic, and implementational — provide a useful framework. The *computational level* asks "what does the system compute and why?" The *algorithmic level* asks "how does it compute it? What representations and processes are used?" The *implementational level* asks "how are the algorithms physically realised?"

Mechanistic interpretability operates primarily at the algorithmic level: it asks what representations (features) and processes (circuits) the network uses to implement its computation. The theory of this monograph formalises this question precisely:

- The computational level corresponds to the task $\mathcal{T}$ and the information-theoretic structure of $p(X, Y)$ (Chapters 1 and 4).
- The algorithmic level corresponds to the circuit $\mathcal{C}$ and the feature frame F (Chapters 1, 2, 5, and 6).
- The implementational level corresponds to the specific weight matrices W and the training dynamics that produced them (Chapters 3 and 9).

The universality theory of Chapter 7 corresponds to the claim that the algorithmic level is largely invariant to the implementational level: different networks implementing the same computation at the computational level will implement the same algorithms at the algorithmic level (the same circuits and features), even if their weights differ at the implementational level.

10.9.2 Explanatory vs. Predictive Interpretability

There are two distinct goals for mechanistic interpretability, which the theory distinguishes sharply.

Explanatory interpretability aims to understand why the network behaves as it does — to identify the circuits and features that are causally responsible for specific outputs. This is the goal of studies like [64]: not just to find a faithful circuit but to identify the circuit that is *causally* responsible for the IOI capability. The theory of Chapter 4 (specifically, Definition 10.5 and Theorem 10.6) formalises when a circuit is causally responsible.

Predictive interpretability aims to use the identified circuits to predict the network's behaviour on novel inputs or tasks — to extrapolate from the identified algorithm to new settings. This is harder, because it requires not just identifying the circuit but verifying that the circuit's behaviour generalises beyond the distribution used for circuit discovery. The statistical limits of Chapter 8 (Theorem 8.5) quantify how hard this is.

The theory draws a sharp line between these two goals: explanatory interpretability is achievable with polynomial samples under the faithfulness criterion (2.1), while predictive interpretability (over all distributions within TV-distance ρ of the training distribution) requires exponentially many samples in the dimension of the input.

10.9.3 The Hypothesis-Driven Approach

The statistical limit of Theorem 8.5 might seem to make predictive interpretability hopeless. But the bound is a worst-case bound over all possible distribution shifts. In practice, the distribution shifts of interest are not adversarially chosen but are constrained by the structure of the task and the domain.

The practical resolution is the *hypothesis-driven approach*: rather than trying to verify the circuit's behaviour on all distributions, one formulates specific hypotheses about how the circuit should behave under specific, structured interventions, and tests those hypotheses empirically. For example:

- Hypothesis: "The name mover heads move information about the indirect object because they attend to it, not because of some confounding factor." Test: patch the attention weights to make the NMH attend to the subject instead, and verify that the output changes to the subject name.
- Hypothesis: "The S-inhibition heads suppress the subject's token embedding by writing a negative signal to its position." Test: remove the SIH from the circuit and verify that the network now predicts the subject's name rather than the indirect object's name.

These hypotheses correspond to causal faithfulness claims (Definition 10.5) and can be tested with polynomial samples. The hypothesis-driven approach thus sidesteps the worst-case sample complexity bound by restricting to structured, domain-specific distribution shifts.

10.10 A Research Agenda

We close by proposing a concrete research agenda that follows from the theory.

Short-term (1–2 years).

(1) Systematically measure the feature importance distributions λ_i in large language models and verify the power-law hypothesis (Proposition 9.1) across multiple architectures and training regimes.

(2) Implement the ℓ^1 relaxation (2.3) at scale, using differentiable masking, and apply it to identify circuits for 10–20 capabilities across model sizes.

(3) Test the phase boundary prediction (Theorem 6.4) empirically by training models with varying sparsity constraints and measuring the number of monosemantic features as a function of sparsity.

Medium-term (2–5 years).

(1) Prove the universality theorem (Theorem 7.4) under realistic training conditions (without assuming IB optimality). This is Problem 3 of Appendix B.

(2) Develop scalable algorithms for testing causal faithfulness of large circuits. This is Problem 5 of Appendix B.

(3) Connect the feature geometry to moral philosophy by identifying value features, measuring their inner products, and comparing to the philosophical structure of values (Hypothesis 10.9).

Long-term (5+ years).

(1) Develop a complete mathematical theory of mechanistic interpretability that makes the heuristic arguments of this monograph (specifically, Theorems 3.3 and 9.2) fully rigorous.

(2) Characterise the class of tasks for which the minimal faithful circuit is mechanistically irreducible (Theorem 8.7), and determine whether any capabilities of large language models fall in this class.

(3) Develop a formal theory of alignment as circuit correspondence (Definition 10.7) and identify training procedures that provably produce aligned networks.

Bibliographic Notes

Marr's three levels of analysis are from [42]. On AI safety and deceptive alignment: [33]. On discovering latent knowledge: [14] and [70]. On concrete problems in AI safety: [3]. On scalable oversight: [18]. On steering vectors: [59] and [63]. Pearl's do-calculus: [50].

10.11 Mechanistic Red-Teaming

Traditional red-teaming for AI safety involves finding inputs that cause a model to produce undesirable outputs. *Mechanistic red-teaming* extends this by identifying the circuits responsible for specific failure modes, enabling more targeted and comprehensive safety analysis.

Definition 10.10 (Mechanistic Red-Team)**.** A *mechanistic red-team* for capability g in network $\mathcal{N}$ is a triple $(\mathcal{C}_g, x_{\text{adv}}, \mathcal{I})$ where:

(1) $\mathcal{C}_g \subseteq \mathcal{N}$ is a faithful circuit for the capability g;

(2) x_{adv} is an adversarial input designed to elicit capability g on a deployment input where g is undesirable;

(3) $\mathcal{I}$ is a set of interventions (ablations, steering vectors, activation patches) that suppress the circuit $\mathcal{C}_g$ and reduce the likelihood of g being activated.

The mechanistic red-team is more informative than a standard red-team because it identifies the circuit responsible for the failure mode (C_g), *theinputconditionsthattriggerit*$(\mathrm{x}_{\text{adv}})$, and potential mitigations $(\mathcal{I})$.

Proposition 10.11 (Mechanistic Red-Teaming and Circuit Discovery)**.** *A mechanistic red-team $(\mathcal{C}_g, x_{\text{adv}}, \mathcal{I})$ for capability g exists if and only if:*

(1) There exists a circuit $\mathcal{C}_g$ that is (ε, D)-faithful for g;

(2) x_{adv} lies in the support of D but outside the intended deployment distribution D_{safe};

(3) The interventions $\mathcal{I}$ are causally faithful (Definition 10.5) for suppressing g.

Proposition 10.11 says that mechanistic red-teaming is possible whenever the dangerous capability is implemented by an identifiable circuit. If the circuit is not identifiable (e.g., because it is mechanistically irreducible, Definition 8.6), mechanistic red-teaming fails, and one must fall back to behavioural red-teaming.

10.11.1 Circuit-Based Mitigation Strategies

Once a mechanistic red-team has identified the circuit $\mathcal{C}_g$ responsible for a dangerous capability, several mitigation strategies are available.

Ablation. Set the weights of the edges in $\mathcal{C}_g$ to zero. This is the most direct mitigation but may degrade other capabilities if $\mathcal{C}_g$ participates in multiple tasks (polyfunctionality, §2.9).

Steering. Apply a steering vector (an activation vector that moves the representation away from the region where $\mathcal{C}_g$ activates). This is less disruptive than ablation but requires identifying the correct steering direction, which is the feature direction f_g corresponding to the dangerous capability.

Fine-tuning with circuit awareness. Fine-tune the model on a dataset designed to reduce the activation of $\mathcal{C}_g$ on harmful inputs, while preserving its activation on benign inputs. The circuit framework provides a principled criterion for designing such a dataset: examples where $\mathcal{C}_g$ activates should be rare and penalised.

Architectural intervention. Modify the architecture to prevent the formation of $\mathcal{C}_g$ during training. For example, if $\mathcal{C}_g$ requires K-composition between two specific heads (high $c_K(\text{head}_1, \text{head}_2)$), one can add a regularisation term penalising high composition scores between those heads.

10.12 The Neuroscience–AI Interpretability Bridge

A recurring theme throughout this monograph is the connection between mechanistic interpretability and neuroscience. The two fields are studying related problems — understanding the internal organisation of information processing systems — using related mathematical tools. We explicitly develop this bridge.

10.12.1 Shared Mathematical Tools

The mathematical tools used in mechanistic interpretability are largely borrowed from computational neuroscience:

- *Population codes and superposition* (Chapter 1): directly from Seung and Sompolinsky [56].
- *Associative memory and Hopfield networks* (Chapter 5): from Hopfield (1982) [32] and Amit et al. [2].
- *Sparse coding* (§1.9): from Olshausen and Field (1996).
- *Representation Similarity Analysis* (Chapter 7): from Kriegeskorte et al. [38].

The flow of mathematical tools has been bidirectional: ideas from neuroscience have been applied to AI (as above), and ideas from AI have been applied back to neuroscience (e.g., deep learning models of the visual cortex, reviewed in [69]).

10.12.2 Shared Empirical Findings

Several empirical findings in mechanistic interpretability have direct parallels in neuroscience:

- *Polysemanticity*: Individual neurons in both artificial and biological networks respond to multiple stimuli. In the primate visual cortex, individual neurons in IT cortex are polysemantic [69]; in language models, individual neurons are polysemantic [24].
- *Sparse activations*: Both biological and artificial networks use sparse activations. In the hippocampus, only ~ 2–5% of neurons are active at any time [48]; in language models, the fraction of active features is estimated to be ~ 1–10%.
- *Modular organisation*: Both biological and artificial networks are modularly organised, with identifiable sub-systems responsible for specific functions. In the brain, this is the organisation of cortical areas; in transformers, this is the organisation of attention heads.

The convergence of these empirical findings across systems as different as biological brains and artificial transformers is striking. The theory of this monograph suggests an explanation: these are not coincidences but optimal solutions to the same representational problem, as predicted by the universality theory of Chapter 7.

Bibliographic Notes

On AI safety: [33], [14], [70], [3], [18], [59], [63]. Pearl's do-calculus: [50]. Marr's levels: [42]. Population codes: [56]. Hopfield networks: [32], [2]. Sparse coding: Olshausen and Field (1996). RSA: [38]. Deep learning models of cortex: [69].

10.13 Interpretability-Based Anomaly Detection

The circuit framework provides a principled approach to *anomaly detection*: identifying inputs on which the network is behaving unusually, potentially indicating a failure mode or an out-of-distribution input.

10.13.1 Circuit Activation Anomalies

For a faithful circuit $\mathcal{C} \subseteq \mathcal{N}$ for task $\mathcal{T}$, define the *circuit activation profile* as the vector of edge attributions $(\mathrm{MI}(e))_{e \in E^{\mathcal{C}}}$. Under the training distribution D, this profile has a characteristic distribution: most edges have attribution scores close to their training-set averages, with fluctuations bounded by the sampling error.

An *anomalous input* is one for which the circuit activation profile differs significantly from its training-distribution average:

$$\mathrm{AS}(x) \;=\; \left\| (\mathrm{MI}(e, x))_{e \in E^{\mathcal{C}}} - \mathbb{E}_{x' \sim D}[(\mathrm{MI}(e, x'))_{e \in E^{\mathcal{C}}}] \right\|_2 , \tag{10.3}$$

where $\mathrm{MI}(e, x)$ is the edge attribution score on specific input x (computed by activation patching on x). A large anomaly score indicates that the circuit is activating in an unusual pattern, which may indicate:

- An out-of-distribution input (the circuit is being asked to process something it was not designed for);
- An adversarial input (the circuit is being manipulated to produce incorrect output);
- A novel task (the circuit is being applied to a task it does not implement faithfully).

10.13.2 Feature-Level Anomaly Detection

At a coarser level, anomaly detection can be performed using the feature importance scores $\lambda_i = I(f_i^\top T; Y)$. On a typical input $x \sim D$, each feature i is activated with probability p_i and the average activation is $\mathbb{E}[a_i(x)] = p_i$. An anomalous input is one where the activation pattern $(a_i(x))_{i=1}^m$ differs significantly from its expected value:

$$\text{FAS}(x) \;=\; \sum_{i=1}^{m} \lambda_i \left| a_i(x) - p_i \right|^2, \tag{10.4}$$

the weighted squared deviation of feature activations from their expected values.

The feature anomaly score (10.4) is computable without running the full circuit discovery algorithm — it requires only the feature directions f_i and the activation values $a_i(x) = f_i^\top h(x)$ — making it efficient for large-scale deployment.

Proposition 10.12 (Feature Anomaly and Distribution Shift)**.** *Let D be the training distribution and D' be a shifted distribution with TV-distance $\text{TV}(D, D') = \rho$. Then:*

$$\mathbb{E}_{x \sim D'}[\text{FAS}(x)] \;\geq\; \rho \cdot \sum_i \lambda_i \cdot \max\left(0,\, \frac{(p_i^{D'} - p_i^D)^2}{p_i^D(1 - p_i^D)}\right), \tag{10.5}$$

where p_i^D and $p_i^{D'}$ are the activation probabilities of feature i under D and D'. The expected feature anomaly score is positive whenever there is distribution shift ($\rho > 0$), with the magnitude proportional to the shift and the importance of affected features.

This proposition says that feature anomaly detection is sensitive to distribution shifts that affect important features. Shifts in low-importance features (small λ_i) are not detected; shifts in high-importance features (large λ_i) are detected with signal proportional to $\lambda_i \rho$.

Bibliographic Notes

AI safety: [33], [14], [70], [3], [18]. Steering vectors: [59], [63]. Pearl: [50]. Marr: [42]. Anomaly detection in the context of neural network safety is studied in Hendrycks and Gimpel (2017), "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks."

Chapter 11

Synthesis: The Mathematical Architecture of Understanding

What we cannot speak about we must pass over in silence.

Ludwig Wittgenstein, *Tractatus Logico-Philosophicus*

The preceding ten chapters have developed a mathematical theory of mechanistic interpretability from its foundations in frame theory (Chapter 1) to its applications in AI safety (Chapter 10). This final chapter synthesises the main themes, traces the connections between them, and reflects on what the theory has achieved and what remains to be done.

11.1 The Three Fundamental Dualities

The theory is organised around three fundamental dualities, each connecting two seemingly separate mathematical structures.

11.1.1 The Feature–Circuit Duality

The first and most central duality is between *features* and *circuits*, established in Theorem 4.4 of Chapter 4. Features are the representational atoms: unit vectors in activation space that the network uses to encode properties of its inputs. Circuits are the computational molecules: sparse subgraphs of the computational graph that implement identifiable functions by composing features. The duality says that neither is primary:

- A feature without a circuit to compute it is inert. It may exist as a direction in weight space, but it carries no information until a circuit activates it on appropriate inputs.
- A circuit without well-separated features to manipulate is uninterpretable. The circuit's computation may be faithful (accounting for the network's behaviour) but not readable — its intermediate representations have no stable correspondence with identifiable properties of the input.

The feature–circuit duality is a *semantic* duality: it connects the *meaning* (features, representations) to the *syntax* (circuits, computations). A mechanistic account of a neural network computation must specify both sides of this duality.

11.1.2 The Geometry–Information Duality

The second duality is between the *geometry of feature space* and the *information theory of representations*. The Welch bound (Theorem 1.1) establishes a constraint on how many

features can coexist in $\mathbb{R}^d$ with low pairwise interference. The rate-distortion lower bound (Lemma 4.1) establishes a constraint on how much information a representation of dimension d can carry about a label Y.

These two constraints are not independent. Corollary 4.11 shows that the geometric condition of incoherence (small coherence $\mu(F)$) is equivalent to the information-theoretic condition of minimality (the representation stores no redundant information). The Welch bound and the rate-distortion bound are two faces of the same underlying constraint: in a d-dimensional space, you cannot simultaneously have many features, high accuracy, and no redundancy. Something must give.

11.1.3 The Dynamics–Complexity Duality

The third duality is between the *training dynamics* of circuit emergence and the *computational complexity* of the task. Theorem 3.3 (Chapter 3) establishes that grokking is a thermodynamic phase transition driven by weight decay, with the transition time determined by the norm gap (Lemma 3.2) between the memorisation solution and the algorithmic solution. Proposition 4.7 (Chapter 4) establishes that the size of the minimal faithful circuit is bounded by the Kolmogorov complexity of the task.

The duality is: the norm gap is the thermodynamic signature of the Kolmogorov complexity gap. A task with low Kolmogorov complexity (short algorithm) has a small algorithmic circuit (small $\|W_{\text{alg}}\|_F^2$) and a large norm gap (large $\|W_{\text{mem}}\|_F^2 - \|W_{\text{alg}}\|_F^2$), making weight decay very effective at driving grokking. A task with high Kolmogorov complexity has a large algorithmic circuit (large $\|W_{\text{alg}}\|_F^2$) and a smaller norm gap, making grokking slower or impossible.

The three dualities are connected: the feature–circuit duality says what we are studying; the geometry–information duality says what the constraints are; and the dynamics–complexity duality says how the system evolves toward the optimal solution under those constraints.

11.2 The Interpretability Landscape

The interpretability–performance frontier (Definition 8.8 and Theorem 8.9) defines the landscape of achievable interpretability for any task. We can now give a more complete description of this landscape using all the results developed in the monograph.

11.2.1 The Axes of the Frontier

The IPF has two axes:

- The *performance axis* ε: the test error of the network. Performance is bounded below by the irreducible entropy $\mathcal{L}_\infty$ (the unavoidable randomness in the label Y).

- The *interpretability axis* K_0: the Kolmogorov complexity budget for the circuit. Interpretability is bounded below by the Kolmogorov complexity K^* of the task (the minimum complexity of any faithful circuit).

The IPF (Theorem 8.9) says that the frontier is convex, monotone, and asymptotically approaches the point $(\mathcal{L}_\infty, K^*)$ as the performance improves. The practical interpretability of a network corresponds to a point on this frontier: a specific tradeoff between how well the network performs (ε) and how simply its computation can be described (K_0).

11.2.2 Where Models Sit on the Frontier

Different types of models sit at different points on the IPF:

- *Small, trained models* (e.g., GPT-2 small): near the right side of the frontier (moderate ε, small K_0). These models have discovered the high-importance circuits (large λ_i) but miss many low-importance features. Their circuits are interpretable (small K_0) but their performance is limited.

- *Large, frontier models* (e.g., GPT-4 class): near the bottom of the frontier (small ε, large K_0). These models have discovered many more features and circuits, achieving near-optimal performance, but the sheer number of circuits (Proposition 9.3) makes comprehensive interpretation infeasible.

- *Interpretability-constrained models*: trained with VIB regularisation (Proposition 4.9) or sparsity constraints to move toward Phase IV of the phase diagram (Chapter 6). These models trade some performance for interpretability, sitting at intermediate points on the frontier.

The interpretability research agenda can be understood as an effort to characterise the frontier — to understand which points are achievable, what circuits correspond to each point, and how to train models that sit at desirable points.

11.3 The Role of Sparsity in the Theory

Sparsity is the thread that runs through every chapter of this monograph. It is not merely a computational convenience but a fundamental feature of the representational problem.

In Chapter 1, sparsity is the condition that makes superposition feasible: if at most k features are active simultaneously, the read-out interference is bounded by $k\varepsilon^2$, which can be made small even when $m \gg d$. Without sparsity, superposition is incoherent.

In Chapter 2, sparsity is the circuit structure that makes circuits interpretable: a sparse circuit (small $|E^{\mathcal{C}}|$) is easier to describe and understand than a dense one. The ℓ^1 relaxation (2.3) promotes sparsity in the circuit structure, and the RIP (Definition 1.3) guarantees that sparse circuits can be recovered exactly.

In Chapter 3, sparsity is the property of the algorithmic solution that makes grokking possible: the algorithmic circuit is sparse ($s \ll N$), giving it a norm advantage (Lemma 3.2) over the memorisation solution and making it thermodynamically preferred under weight decay.

In Chapter 6, sparsity is the control parameter that drives the splitting transition: increasing sparsity (the activation parameter k) triggers feature splitting (Theorem 6.4), moving the network from Phase I (dense-polysemantic) to Phase IV (sparse-monosemantic).

In Chapter 9, sparsity appears as the activation density of the feature representation, which determines the interference level and hence the number of features a model of N parameters can represent (Proposition 9.3).

The central role of sparsity across all chapters suggests that a unified theory of mechanistic interpretability must be fundamentally a theory of *sparse representations and sparse computations*. The mathematical language of this theory — frames, RIP, compressed sensing — is the language of sparsity.

11.4 Connections to Adjacent Fields

The theory developed here connects to several adjacent fields in ways that deserve explicit mention.

11.4.1 Theoretical Machine Learning

The circuit framework is closely related to the *computational learning theory* of neural networks [4]. The VC dimension, Rademacher complexity, and PAC learning framework all measure aspects of a network's capacity that are related to the number and complexity of its circuits. The specific contribution of this monograph is to make the connection to *interpretability* explicit: not just how much a network can learn, but *how* it learns, in terms of identifiable circuits and features.

The depth-width tradeoff (Theorem 2.7) connects to the classical results on the computational power of depth in Boolean circuits [4]. The Kolmogorov complexity bounds (Propositions 4.7 and 4.7) connect to the MDL literature in statistics and information theory [29].

11.4.2 Statistical Mechanics

The phase transition framework of Chapters 3 and 6 connects directly to the statistical mechanics of disordered systems. The free energy argument of §3.5 is a direct application of Kramers' escape rate formula [30], and the symmetry-breaking analysis of §6.6 uses the standard Landau theory of continuous phase transitions [27].

The connection is not merely metaphorical. The training dynamics of neural networks can be rigorously mapped to the dynamics of a statistical mechanics system, with the weight decay coefficient playing the role of temperature and the network weights playing the role of the order parameter. The grokking phenomenon is, in this language, a first-order phase transition in which the system tunnels from the memorisation basin to the algorithmic basin.

The modern Hopfield network connection (Chapter 5) is the most direct link to statistical mechanics: the Hopfield network was originally introduced as a physical model of associative memory in the brain, and its capacity analysis [2] uses the replica method from spin glass theory. The modern version [53] connects to the Gibbs distribution and free energy minimisation.

11.4.3 Harmonic Analysis

The frame theory of Chapter 1 is a branch of harmonic analysis, specifically the theory of redundant bases for function spaces [17]. The Welch bound is a consequence of the Gram matrix analysis, which is a standard tool in harmonic analysis.

The Fourier analysis of grokking (§3.4) connects to the classical theory of characters of finite abelian groups, which is the discrete version of Fourier analysis. The observation that the grokked circuit for modular addition implements the Fourier algorithm is a concrete example of the general principle that circuits implementing tasks with group symmetry will have Fourier structure.

11.4.4 Algebraic Geometry and Representation Theory

The classification of optimal feature configurations (Proposition 6.6) connects to the theory of regular polytopes in algebraic geometry. The moduli space of UNTFs (§6.7) is a complex algebraic variety whose topology encodes the structure of the feature splitting transitions.

The universality theory of Chapter 7 connects to the representation theory of symmetry groups: the symmetry group of the task $p(X, Y)$ acts on the space of optimal bottleneck

representations, and the universality class is the orbit of the representation under this group action.

11.5 What the Theory Achieves

Looking back over the ten preceding chapters, it is worth stating explicitly what the theory has and has not achieved.

11.5.1 What Has Been Achieved

(1) *Precise definitions*: features (unit vectors in activation space), circuits (subgraphs of the computational graph), faithfulness (Definition 2.4), interpretability (Corollary 4.8). These definitions make previously vague concepts mathematically precise.

(2) *Fundamental bounds*: the Welch bound on interference, the RIP-based capacity bound, the rate-distortion lower bound, the Kolmogorov complexity bound on circuit size. These bounds are tight in various senses and cannot be improved without changing the problem.

(3) *Phase diagram*: the (λ, k) phase diagram of feature configurations (Chapter 6) with four qualitatively distinct phases and a computable phase boundary (Theorem 6.5). This phase diagram makes the phenomenology of feature splitting and merging predictable.

(4) *Connection to scaling laws*: the derivation of the Chinchilla exponents from the power-law feature importance distribution (Theorem 9.2). This is the first principled derivation of the scaling law exponents from microscopic principles.

(5) *Universality theory*: the prediction (Theorem 7.4 and Hypothesis 7.3) that independently trained networks implement circuits in the same universality class. This is a testable prediction that has partial empirical support [34, 46].

(6) *Fundamental limits*: the undecidability of exact circuit verification (Theorem 8.1), the Σ_2^P-hardness of approximate circuit discovery (Theorem 8.3), and the shape of the interpretability–performance frontier (Theorem 8.9). These limits define the boundaries of what any interpretability method can achieve.

11.5.2 What Has Not Been Achieved

(1) *A rigorous proof of the universality theorem*: Theorem 7.4 assumes information bottleneck optimality, which is not guaranteed by gradient descent. The theorem provides the right qualitative prediction but lacks a rigorous derivation under realistic training conditions.

(2) *An efficient circuit discovery algorithm*: the ℓ^1 relaxation (2.3) is tractable but achieves only an $O(\log n)$ approximation ratio (Proposition 8.4). Better algorithms are known to be impossible under standard complexity-theoretic assumptions, but domain-specific algorithms that exploit the structure of neural circuits might do better.

(3) *A theory of feature semantics*: the theory describes the geometry of features (their mutual angles, norms, and frames) but not their content. It says nothing about *which* features a network will learn, only about the geometric structure of the feature space given a fixed set of features. A theory of feature semantics — connecting the geometry of the feature space to the statistical structure of the input distribution — would be a significant advance.

(4) *A complete treatment of non-linear circuits*: the analysis of superposition in Chapter 1 assumes that features are read out by linear projections. Non-linear circuits (involving polynomial interactions between features, as in attention-based computation) require a more general framework. Chapter 5 treats the attention mechanism specifically, but a general theory of non-linear circuits remains to be developed.

11.6 A Final Reflection

The title of this monograph — *Circuits, Features, and the Geometry of Meaning* — contains a word that deserves, at the end, some explicit attention: *meaning*.

The features that mechanistic interpretability identifies are not merely statistical patterns in activation space. They carry semantic content: the "days of the week" feature means something about days of the week; the "pronoun" feature means something about pronouns; the "harmlessness" feature means something about moral valence. The circuits that compose these features implement algorithms that reason about concepts, not merely transform vectors.

The theory developed here treats features as geometric objects (unit vectors in $\mathbb{R}^d$) and circuits as combinatorial objects (subgraphs of a DAG). It says nothing, directly, about meaning. This is a deliberate abstraction: the mathematical theory is developed at a level that is agnostic about the semantic content of the features, focusing instead on their geometric and computational structure.

But the abstraction is also a limitation. A complete theory of mechanistic interpretability would need to connect the geometric structure of features to their semantic content — to explain not just that a network represents a binary property as a direction in $\mathbb{R}^d$, but *why* that direction encodes that property, and *how* the training process selects that encoding from the space of all possible encodings.

This connection — from geometry to semantics, from circuits to meaning — is the deepest open problem in the field. It is, in some sense, the problem that the title of this monograph gestures at but does not solve. The geometry of meaning awaits its theory.

Bibliographic Notes

The three dualities discussed in §11.1 synthesise results from Chapters 1–4. The IPF landscape of §11.2 combines Theorem 8.9 with the phase diagram of Chapter 6. The role of sparsity across chapters follows the thread established in §§1.3, 2.4, 3.3, 6.4, and 9.3. Connections to theoretical machine learning draw on [4]. Statistical mechanics connections draw on [30], [27], and [2]. Frame theory connections are in [17]. The Wittgenstein epigraph is from *Tractatus Logico-Philosophicus* (1921).

11.7 The Future of the Theory: Open Directions

Beyond the specific open problems listed in Appendix B, there are several broader directions in which the theory can develop.

11.7.1 A Theory of Feature Semantics

The theory developed in this monograph is purely structural: it characterises the geometry of feature space and the topology of circuit structure, without saying anything about the semantic content of features. A feature is, in this theory, simply a unit vector $f_i \in \mathbb{R}^d$; its meaning (what property of the input it encodes) is entirely implicit, derived from the training process and the distribution $p(X, Y)$.

A theory of feature semantics would connect the geometry of the feature space to the statistical structure of the input distribution. The key question is: given $p(X, Y)$, what features will the network learn? The answer must involve both the information-theoretic structure of $p(X, Y)$ (which features are most informative about Y) and the geometric structure of the feature space (which features can coexist with minimal interference).

A natural conjecture, suggested by the sufficient statistics analysis of Chapter 4, is that the network learns the *natural statistics* of $p(X, Y)$: the directions in input space that are most predictive of Y, in decreasing order of predictiveness. These are the principal components of the Fisher information matrix of $p(Y|X)$, which are the "natural" features for the prediction task.

11.7.2 The Dynamics of Feature Learning

Related to the semantics question is the question of *how* features are learned. The grokking analysis of Chapter 3 describes the emergence of a fixed algorithmic circuit, but it assumes the circuit exists and is waiting to be discovered. A more complete theory would describe the *discovery process*: how gradient descent, starting from a random initialisation, identifies the relevant features and builds circuits connecting them.

The phase transition framework suggests a scenario: early in training, many features are learned simultaneously (the memorisation phase), in a high-interference configuration. As training proceeds, weight decay creates pressure toward lower-norm solutions, and the features gradually separate (the latent circuit formation phase), approaching the Welch-bound geometry. At grokking, the circuit crystallises. After grokking, the feature frame approaches the optimal geometry (cross-polytope or simplex, depending on the number of features relative to the dimension).

This scenario can be formalised as a *diffusion process on the space of frame configurations*: the frame $F_t = \{f_1^{(t)}, \ldots, f_m^{(t)}\}$ evolves under the gradient flow of the frame interference energy $\mathcal{E}(F)$ perturbed by the gradient of the task loss. The stationary distribution of this diffusion is the Gibbs distribution $\propto e^{-\beta_{\text{eff}}\mathcal{E}(F)}$ where β_{eff} is the ratio of the regularisation strength to the learning rate.

11.7.3 Non-Linear Extensions

The superposition analysis of Chapter 1 assumes that features are read out by linear projections $f_i^\top h$. Real neural networks use non-linear read-outs (ReLU activations, attention softmax, etc.), and the theory needs to be extended to handle these.

The natural extension is to replace the linear feature frame $\{f_i\}$ with a *non-linear feature dictionary*: a collection of functions $\phi_i : \mathbb{R}^d \to \mathbb{R}$ (one per feature), such that the activation of feature i on representation h is $a_i(h) = \phi_i(h)$. The interference between features i and j is then the expected squared cross-activation: $\mathbb{E}_h[(\phi_i(h) - \mathbb{E}[\phi_i(h)])(\phi_j(h) - \mathbb{E}[\phi_j(h)])]$, which reduces to the inner product $\langle f_i, f_j \rangle$ in the linear case.

For non-linear features, the Welch bound (Theorem 1.1) no longer applies directly, but analogous results can be derived using the theory of *random feature maps* and the kernel methods framework. The capacity-interference tradeoff (Proposition 1.7) generalises to a bound on the kernel matrix of the feature dictionary, which replaces the Gram matrix G of the frame.

11.7.4 The Theory in the Age of Foundation Models

The theory developed here was motivated by, and tested against, interpretability findings on relatively small models (GPT-2 scale, toy models of superposition, small transformers for

modular arithmetic). The question of whether it extends to *foundation models* — models of 10^{11} to 10^{12} parameters — is both empirically open and theoretically important.

The scaling law analysis of Chapter 9 gives some predictions: at scale, the number of features $m^*(N)$ grows polynomially in N, the feature interference $\varepsilon \sim N^{-1/4}$ decreases, and the circuit count $c^*(N)$ grows with a specific exponent determined by γ. These are quantitative predictions that can be tested as interpretability tools improve.

The universality theory of Chapter 7 predicts that the circuits for high-importance capabilities are universal across model sizes: the IOI circuit of GPT-2 should have the same universality class as the IOI circuit of GPT-4 (if one exists at comparable abstraction level). Testing this prediction requires applying the same circuit discovery methods to models of very different sizes and checking whether the discovered circuits are $(O(\varepsilon), O(\varepsilon))$-isomorphic (Definition 7.1).

11.8 Philosophical Coda: What It Means to Understand a Mind

We began this monograph with a scientific crisis — the opacity of neural networks — and proposed a mathematical theory as the response. We end with a philosophical observation about what kind of understanding the theory provides and what kind it does not.

The theory of this monograph is a *structural* understanding: it characterises the circuits and features of a neural network as mathematical objects, with precise geometric and information-theoretic properties. This structural understanding is genuine and non-trivial — it is more than a description of the network's input-output behaviour, because it characterises the network's internal representations and their organisation.

But structural understanding is not the only kind of understanding, and it may not be the kind that matters most for safety. What we ultimately want to understand about a neural network is not its structure but its *intentions*: what goals it is pursuing, what values it is optimising for, how it will behave in novel situations where its training distribution does not apply.

The theory approaches this question through the alignment criterion (Definition 10.7) and the geometry of values (Hypothesis 10.9). But these approaches are at best partial: they tell us what features encode value-related properties and how they interact, but they do not tell us whether the system is genuinely pursuing those values or merely representing them as correlates of its training signal.

This gap — between representing values and pursuing them — is the deepest challenge in AI alignment. It is not a gap that the mathematical theory of this monograph can close, because it is not a mathematical gap but a philosophical one: the question of what distinguishes a system that is genuinely pursuing a goal from one that merely represents a goal-related feature is a question about the nature of agency, intentionality, and mental content.

The theory can, however, help sharpen the question. By identifying the features that encode value-related properties and characterising their causal role in the network's computation (Theorem 10.6), the theory can determine whether a given value feature is causally responsible for the network's behaviour or merely correlated with it. The former is a necessary (though not sufficient) condition for the network to be genuinely pursuing the corresponding value.

Whether it is sufficient — whether causal responsibility for value-related features is the right notion of "genuinely pursuing a value" for an artificial system — is a question that lies at the intersection of the theory of this monograph and the philosophy of mind. It is a question we leave open, with the observation that the mathematical theory, by making the question

precise, has at least made it possible to ask it in a way that admits, in principle, a definite answer.

Bibliographic Notes

The three dualities of §11.1 synthesise results from Chapters 1–4. The IPF landscape is from Theorem 8.9. The sparsity thread synthesises §§1.3, 2.4, 3.3, 6.4, and 9.3. The information geometry of circuits connects to [1]. The philosophical discussion in §11.6 connects to the philosophy of mind literature, particularly Searle (1980) on intentionality and the Chinese Room, and Dennett (1987) on the intentional stance. The Wittgenstein epigraph is from the *Tractatus Logico-Philosophicus* (Proposition 7, 1921).

11.9 Bridges to Classical Mathematics

The theory developed in this monograph sits at the intersection of several classical mathematical disciplines. We now trace these connections more explicitly, both to situate the work and to suggest where classical results might be imported to sharpen the theory.

11.9.1 Convex Geometry and the Geometry of Features

The geometry of feature configurations (Chapter 1) is a problem in *convex geometry* and *sphere packing*. The question of how many nearly-orthogonal unit vectors can coexist in $\mathbb{R}^d$ (Lemma 1.6) is the sphere packing problem on S^{d-1}: how many spherical caps of angular radius $\theta = \arccos(\varepsilon)$ can be placed on S^{d-1} without overlapping?

The classical Cohn–Elkies linear programming bound [20] gives upper bounds on the density of sphere packings that are tight in dimensions 8 and 24 (the E_8 and Leech lattice configurations). For the mechanistic interpretability problem, these bounds translate into upper bounds on the number of features m that can be represented with interference ε in dimension d.

The Welch bound (Theorem 1.1) is the linear programming bound in the special case where all pairwise overlaps are equal (the equiangular case). For non-equiangular configurations, the Cohn–Elkies approach gives tighter bounds that account for the distribution of pairwise overlaps.

Remark 11.1 (The E_8 Lattice and Feature Geometry). In dimension $d = 8$, the E_8 lattice achieves the sphere packing density upper bound. Applied to the feature geometry problem, this means that in a representation space of dimension $d = 8$, the optimal configuration of features (minimising the maximum coherence) for specific values of m is related to the E_8 root system. Whether neural networks trained in low-dimensional settings spontaneously discover E_8-like feature configurations is an open empirical question.

11.9.2 Algebraic Topology and Circuit Homotopy

The sequential composition of circuits $\mathcal{C}_2 \circ \mathcal{C}_1$ (Definition 2.2) defines a category of circuits. The universality theory of Chapter 7 can be reformulated in the language of algebraic topology: two circuits are in the same universality class if and only if they are *homotopic* in the space of circuit implementations.

More precisely, define the *circuit space* $\mathcal{C}_\mathcal{T}$ as the set of all faithful circuits for task $\mathcal{T}$, with the topology induced by the Frobenius norm on weights. Two circuits $\mathcal{C}$ and $\mathcal{C}'$ are homotopic if there exists a continuous path $\{\mathcal{C}_t\}_{t\in[0,1]}$ in $\mathcal{C}_\mathcal{T}$ with $\mathcal{C}_0 = \mathcal{C}$ and $\mathcal{C}_1 = \mathcal{C}'$.

The fundamental group $\pi_1(\mathcal{C}_\mathcal{T})$ (the group of homotopy classes of loops in the circuit space) classifies the distinct "types" of circuits for task $\mathcal{T}$. The universality theory predicts that $\pi_1(\mathcal{C}_\mathcal{T})$ is determined by the symmetry group of $p(X, Y)$, and that gradient descent typically discovers circuits in the connected component of $\mathcal{C}_\mathcal{T}$ corresponding to the trivial element of π_1.

11.9.3 Harmonic Analysis on Groups and Grokking

The Fourier analysis of modular arithmetic (§3.4) is a special case of harmonic analysis on finite abelian groups. The general theory of Pontryagin duality gives a systematic framework for analysing tasks with group symmetry: the optimal circuit will always implement the Fourier transform of the group, and the grokking transition corresponds to the network discovering the group's character table.

Proposition 11.2 (Grokking as Character Table Discovery)**.** *Let G be a finite abelian group and $\mathcal{T}_G$ be the task of computing a group operation on G. The minimal faithful circuit for $\mathcal{T}_G$ implements the discrete Fourier transform of G, using $|G|$ Fourier modes. The grokking transition corresponds to the network's weights aligning with the character table of G, which is the optimal basis for representing group elements.*

For $G = \mathbb{Z}/p\mathbb{Z}$ (the cyclic group of order p), the character table is the DFT matrix, and Proposition 11.2 reduces to the modular addition analysis of §3.4. For larger groups (e.g., $G = (\mathbb{Z}/p\mathbb{Z})^n$, the group of n-tuples modulo p), the character table has p^n rows and columns, and the minimal faithful circuit has size $O(n \log p)$ (implementing the fast Fourier transform).

This connection to harmonic analysis on groups suggests that any task with group symmetry will exhibit grokking with the same universal mechanism: the network discovers the group's Fourier structure and implements the group's character table. The time to grokking is controlled by the norm gap (Lemma 3.2), which scales as p^n (size of the training set) versus $O(n \log p)$ (size of the FFT circuit).

11.9.4 Statistical Decision Theory and Optimality

The information bottleneck framework of Chapter 4 is closely related to *statistical decision theory* [10], which provides a more general framework for optimal inference under loss functions.

The rate-distortion function $R(D)$ (defined in (4.1)) is the statistical decision theory analogue of the Bayes risk: it gives the minimum achievable expected loss among all decision rules that use a representation of complexity (rate) at most R. The fundamental theorem of rate-distortion theory states that $R(D)$ is achievable by random codes of length R, and not achievable by any code of length less than R.

In the neural network context, this theorem translates to: the minimal faithful circuit for a task of Kolmogorov complexity K^* has at least $K^*/\log(1/\varepsilon)$ edges (the Shannon lower bound on circuit size, from Proposition 4.7), and this bound is achievable by random initialisation of the circuit weights.

The connection to statistical decision theory also illuminates the nature of the information bottleneck tradeoff (4.4): the Lagrange multiplier β plays the role of a cost-per-bit-of-information, balancing the cost of using information (parameter cost in the circuit) against the benefit (reduction in prediction loss).

11.10 Selected Research Problems in Detail

We expand on three of the open problems from Appendix B that we believe are most tractable and most important.

11.10.1 Problem 1: Deriving the Chinchilla Exponents

The derivation of $\alpha = (\gamma - 1)/2$ in Theorem 9.2 rests on two assumptions: (i) feature importances follow a power law with exponent γ, and (ii) the network achieves the Welch bound interference allocation. Making the derivation rigorous requires establishing both assumptions from first principles.

For assumption (i), the key step is to connect the statistical structure of natural language corpora to the power-law hypothesis. A natural approach: define the "importance" of a concept c in a corpus as the mutual information $I(c; Y)$ between the presence of c in the context and the next-word distribution Y. Show that this mutual information follows a power law with exponent γ that can be measured from the corpus. This is an empirical question about the structure of natural language, addressable by large-scale corpus analysis.

For assumption (ii), the key step is to show that gradient descent on a transformer converges to a solution that approximately achieves the Welch bound. A natural approach: show that the ℓ^2 regularisation (weight decay) penalises interference $\mu(F)$, and that the regularised loss landscape has a unique minimum at the Welch-bound configuration. This is a question about the geometry of the loss landscape, addressable by tools from non-convex optimisation.

11.10.2 Problem 3: Universality Beyond IB Optimality

The universality theorem (Theorem 7.4) assumes that gradient descent achieves the information bottleneck optimum. A weaker assumption, more consistent with empirical observations, is that gradient descent achieves the *empirical information bottleneck optimum*: it minimises the empirical IB functional on the training data, rather than the population IB functional.

Under this weaker assumption, the universality conclusion still holds for the training distribution, but may fail for distributions that differ significantly from the training distribution. The gap between population and empirical universality is characterised by the sample complexity bound of Theorem 8.5: to achieve universality across distributions within TV-distance ρ of the training distribution, one needs $N \geq e^{\Omega(d\rho^2)}$ training examples.

A more tractable step toward Problem 3 is to prove universality for the specific training algorithm of SGD with weight decay, under the Langevin dynamics approximation of §3.7. The Langevin dynamics converges to the Gibbs distribution (3.8), which is determined by the regularised loss $\mathcal{L}(W) + \lambda \|W\|_F^2 / 2$. The universality of the Gibbs distribution (independent of initialisation) is a weaker claim than IB optimality, and may be provable using the theory of ergodic Markov chains.

11.10.3 Problem 5: Causal Faithfulness at Scale

Testing causal faithfulness of features (Definition 10.5) in large language models requires the following algorithmic steps:

(1) *Feature identification*: Use CCS or Representation Engineering [14, 70] to identify candidate feature directions $f_i \in \mathbb{R}^d$ for property g_i.

(2) *Sufficiency testing*: Verify $I(f_i^\top T; g_i(X)) \approx H(g_i(X))$ on a held-out set of labelled examples.

(3) *Bypass circuit search*: Apply the ℓ^1 circuit discovery algorithm (2.3) on inputs where f_i is ablated, and check whether any faithful circuit remains. If yes, a bypass circuit exists and f_i is not causally faithful.

(4) *Intervention verification*: Perform causal interventions (activation patching, steering) on f_i and measure the effect on $g_i(X)$ outcomes. Compare to the observational conditional expectation $\mathbb{E}[g_i(X)|f_i^\top T = v]$.

Steps 1 and 2 are already implemented in the CCS and Representation Engineering literature. Step 3 requires an efficient implementation of the circuit discovery algorithm at scale — the main technical bottleneck. Step 4 is routine given the identified feature direction.

A key simplification for Step 3: instead of running the full ℓ^1 circuit discovery, one can test for bypass circuits by the *direct effect* method — computing the direct effect of each attention head on the output, bypassing all intermediate computations. A head with a large direct effect on $g_i(X)$ that does not pass through f_i is a candidate bypass circuit. This reduces the computational complexity from $O(|E|^2)$ (exhaustive circuit search) to $O(|E|)$ (one forward pass per head).

Bibliographic Notes

The three dualities of §11.1: Chapters 1–4. The IPF: Theorem 8.9. Sparsity synthesis: §§1.3, 2.4, 3.3, 6.4, 9.3. Sphere packing and the Cohn–Elkies bound: Cohn and Elkies (2003), "New Upper Bounds on Sphere Packings." Information geometry: [1]. Statistical decision theory: Berger (1985), *Statistical Decision Theory and Bayesian Analysis*. Harmonic analysis on groups: Terras (1999), *Fourier Analysis on Finite Groups and Applications*. The Wittgenstein epigraph: *Tractatus Logico-Philosophicus*, Proposition 7 (1921).

11.11 The Information-Geometric Perspective on the Theory

Throughout this monograph, we have used two distinct geometric perspectives: the Euclidean geometry of representation spaces (Chapter 1) and the combinatorial geometry of circuits (Chapter 2). A third perspective — *information geometry* [1] — provides a unifying framework for both.

11.11.1 The Fisher-Rao Metric on Representations

The space of probability distributions $\{p_\theta(Y|X) : \theta \in \Theta\}$ parametrised by a neural circuit $\mathcal{C}$ carries a natural Riemannian metric — the *Fisher-Rao metric*:

$$ds^2 \;=\; \sum_{\mu,\nu} \mathcal{I}_{\mu\nu}(\theta)\, d\theta^\mu\, d\theta^\nu, \tag{11.1}$$

where $\mathcal{I}_{\mu\nu}(\theta) = \mathbb{E}[\partial_\mu \log p_\theta(Y|X) \cdot \partial_\nu \log p_\theta(Y|X)]$ is the Fisher information matrix. The geodesic distance between two circuits $\mathcal{C}$ and $\mathcal{C}'$ in this metric measures the distinguishability of their prediction distributions.

Proposition 11.3 (Fisher Information and the Welch Bound)**.** *For a circuit $\mathcal{C}$ with feature frame F of coherence μ, the Fisher information matrix satisfies:*

$$\lambda_{\min}(\mathcal{I}_\mathcal{C}) \;\geq\; (1 - m\mu^2) \cdot \frac{\lambda_i}{d}, \tag{11.2}$$

where λ_i is the importance of the least important feature and d is the representation dimension. For incoherent frames ($\mu \leq 1/\sqrt{d}$), the minimum Fisher information is bounded below by $\lambda_i/2d > 0$.

The proposition says that circuits with incoherent feature frames have positive-definite Fisher information — they are *identifiable*: different parameter configurations produce distinguishably different predictions. This is the information-geometric version of the faithful circuit condition: a circuit that is not identifiable cannot be uniquely determined from data.

11.11.2 The Natural Gradient and Circuit Learning

The natural gradient algorithm, introduced by [1], replaces the Euclidean gradient $\nabla_\theta \mathcal{L}$ with the Fisher-Rao metric-adjusted gradient:

$$\tilde{\nabla}_\theta \mathcal{L} \;=\; \mathcal{I}^{-1}(\theta) \nabla_\theta \mathcal{L}, \tag{11.3}$$

which takes steepest ascent steps in the information-geometric sense rather than the Euclidean sense. For circuits with incoherent feature frames, the natural gradient differs from the Euclidean gradient by a factor approximately equal to the inverse importance weights $1/\lambda_i$, causing the natural gradient to take larger steps for low-importance features and smaller steps for high-importance features.

This behaviour is consistent with the empirical observation that neural networks learn high-importance features faster than low-importance ones (§9.4): the natural gradient preferentially adjusts the parameters most relevant to high-importance features.

11.11.3 Connections to the IB Objective

The information bottleneck objective (4.4) can be written as a geodesic distance minimisation in the information-geometric sense:

$$\mathcal{L}_{\text{IB}}(T) \;=\; D_{\text{KL}}(p(T|X) \| r(T)) - \beta \cdot I(T;Y), \tag{11.4}$$

where $D_{\text{KL}}(p(T|X)\|r(T))$ is the KL divergence from the prior $r(T)$ (the "distance" from the prior in the information-geometric sense), and $I(T;Y)$ is the relevance term. Minimising (11.4) over distributions $p(T|X)$ is equivalent to finding the distribution on the geodesic between the prior $r(T)$ and the posterior $p(T|X,Y)$ (the fully informative representation) at the point parametrised by β.

This geodesic interpretation connects the IB framework to the information-geometric theory of statistical manifolds, providing a precise geometric meaning to the "bottleneck" metaphor: the optimal bottleneck representation is the point on the geodesic between prior and posterior that achieves the best tradeoff between compression (distance from prior) and relevance (distance from posterior).

11.12 A Research Programme for the Theory's Continuation

The theory of this monograph has established the framework but left substantial mathematical work undone. We close by stating, as precisely as possible, the research programme that would complete the theory.

11.12.1 The Five Core Theorems to Be Proved

T1. Rigorous universality: Prove Theorem 7.4 without the IB optimality assumption, replacing it with a weaker assumption about the long-time behaviour of SGD with weight decay. The expected tool: the theory of Markov chain mixing times applied to the Langevin dynamics of §3.7 (Proposition 3.5).

T2. Chinchilla from first principles: Prove Theorem 9.2 rigorously, establishing that (i) the feature importance distribution of natural language satisfies $\lambda_i \propto i^{-\gamma}$ with $\gamma \in (1, 2)$, and (ii) the trained network approximately achieves the Welch bound interference allocation.

T3. Causal faithfulness certificate: Prove a polynomial-time algorithm for deciding whether a given feature is causally faithful for a given property, assuming the feature frame is incoherent (coherence $\mu \leq 1/\sqrt{d}$). The expected tool: Proposition 8.17 plus the direct effect method (§10.5, Problem 5).

T4. Scaling of circuit complexity: Prove that the Kolmogorov complexity of the minimal faithful circuit for natural language modelling grows as $\Theta(N^\alpha)$ for the parameter scaling exponent α, establishing the connection between Theorem 9.2 and Proposition 4.7.

T5. Phase diagram for transformers: Characterise the critical points of the transformer circuit energy landscape (the analogue of Theorem 6.5 for multi-layer transformers with attention), identifying the phase boundaries in terms of composition scores (Definition 2.6).

11.12.2 The Three Core Experiments to Be Run

E1. Feature importance measurement: For a language model at each of several scales ($N = 10^7, 10^8, 10^9, 10^{10}, 10^{11}$), measure the feature importance distribution $\{\lambda_i\}$ using the MINE estimator (§4.7) and test the power-law hypothesis (Proposition 9.1) and the scaling law prediction (Theorem 9.2).

E2. Universality across scales: Apply the CKA measure (Proposition 7.13) to compare representations between models of consecutive sizes and test whether circuits for specific capabilities (IOI, indirect object identification, modular arithmetic) are in the same universality class across scales.

E3. Phase diagram confirmation: Train models with systematically varied sparsity constraints (λ, k) and measure the number of monosemantic features as a function of the control parameters, testing the critical sparsity formula (Theorem 6.4) and the discrete feature spectrum (Corollary 6.7) across architectures and scales.

The programme **T1**–**T5** and **E1**–**E3** would, if completed, transform the heuristic arguments of this monograph into a fully rigorous mathematical theory of mechanistic interpretability, empirically validated across model scales and architectures. It is an ambitious programme for a young field. We end with the observation that the field has the tools, the talent, and — most importantly — the motivation to complete it.

Bibliographic Notes

The three dualities of §11.1 synthesise Chapters 1–4. Information geometry: [1]. Fisher-Rao metric and natural gradient: [1], Chapter 2. Statistical decision theory: [10]. Sphere packing

and the Cohn-Elkies bound: [20]. Harmonic analysis on groups: Terras (1999), *Fourier Analysis on Finite Groups*. Wittgenstein epigraph: *Tractatus*, Proposition 7.

Appendix A

Collected Proofs and Technical Complements

This appendix collects the proofs of results stated without full proof in the main text, together with technical complements that support the main development but would have disrupted its flow. We follow the same numbering as the main text, citing the original theorem or lemma before giving the detailed argument.

A.1 Proofs for Chapter 1

A.1.1 The Welch Bound in Full Generality

Theorem 1.1 was proved in the main text for real frames. We record here the complex analogue, which is needed for the connection to ETFs over $\mathbb{C}$.

Theorem A.1 (Complex Welch Bound)**.** *For any collection of m unit vectors $f_1, \dots, f_m$ in $\mathbb{C}^d$ with $m > d$:*

$$\mu(F) \;\geq\; \sqrt{\frac{m-d}{d(m-1)}}.$$

Proof. The proof is identical to the real case, replacing $\mathbb{R}^d$ with $\mathbb{C}^d$ and $G_{ij} = \langle f_i, f_j \rangle$ with the Hermitian inner product. The Cauchy–Schwarz inequality and the rank constraint apply verbatim. $\square$

A.1.2 Proof of the RIP for Random Frames (Theorem 1.5)

Proof. Fix a k-sparse unit vector $a \in \mathbb{R}^m$ with $\operatorname{supp}(a) = S$, $|S| = k$. The vector $Fa = \sum_{i \in S} a_i f_i$ satisfies:

$$\|Fa\|^2 \;=\; 1 + \sum_{\substack{i,j \in S \\ i \neq j}} a_i a_j \langle f_i,\, f_j \rangle.$$

Each cross term $\langle f_i,\, f_j \rangle$ has mean zero and is subgaussian with parameter $1/d$ (from Lemma A.2). By the Hanson–Wright inequality [54]:

$$\mathbb{P}\left[\left| \sum_{i \neq j \in S} a_i a_j \langle f_i,\, f_j \rangle \right| \geq t \right] \;\leq\; 2 \exp\!\left(-c \min\!\left(\frac{t^2 d}{k^2},\; \frac{td}{k} \right) \right).$$

Setting $t = \delta_k$ and taking $\delta_k = O(\sqrt{k \log(m/k)/d})$ ensures the exponent is at most $-c \log(m/k) \binom{m}{k}$, which after a union bound over all $\binom{m}{k}$ choices of S gives the stated probability. The covering argument over the unit sphere in $\mathbb{R}^k$ (to handle all k-sparse unit vectors, not just those with a fixed support) introduces an additional factor of $(9/\delta_k)^k$, absorbed into the constants. $\square$

A.1.3 Concentration of Measure on the Sphere

Lemma A.2 (Gaussian Tail Bound on the Sphere)**.** *Let f be drawn uniformly from S^{d-1} and $v \in S^{d-1}$ be fixed. Then for $t > 0$:*

$$\mathbb{P}[|\langle f, v\rangle| \geq t] \;\leq\; 2e^{-dt^2/2}. \tag{A.1}$$

Proof. Write $f = g/\|g\|$ where $g \sim \mathcal{N}(0, I_d)$. Then $\langle f, v\rangle = \langle g, v\rangle/\|g\|$. By the symmetry of the Gaussian distribution, $\langle g, v\rangle \sim \mathcal{N}(0,1)$ independently of $\|g\|$. The Gaussian tail bound gives $\mathbb{P}[|\langle g, v\rangle| \geq t] \leq 2e^{-t^2/2}$. Conditioning on $\|g\|$ and using that $\|g\|^2 \geq d/2$ with probability at least $1 - e^{-d/8}$ (by the chi-squared tail bound), we get:

$$\begin{aligned}\mathbb{P}[|\langle f, v\rangle| \geq t] &\leq \mathbb{P}[|\langle g, v\rangle| \geq t\,\|g\|] + \mathbb{P}[\|g\|^2 < d/2] \\ &\leq 2e^{-t^2 d/2} + e^{-d/8} \;\leq\; 2e^{-dt^2/2}\end{aligned}$$

for $t \leq 1/\sqrt{2}$, which covers the regime of interest. □

A.1.4 Optimal Magnitude Allocation (Proposition 1.8)

Proof. With fixed ETF directions $\hat{f}_i$ and magnitudes r_i, the expected squared read-out error for feature i is approximately $r_i^{-2} \cdot p_i \cdot (\text{const})$ (since missing a feature of unit activation costs r_i^{-2}). The total expected loss is:

$$\mathcal{L}(\{r_i\}) \;=\; \sum_{i=1}^{m} \frac{c\,p_i}{r_i^2} \quad \text{subject to} \quad \sum_{i=1}^{m} r_i^2 = E.$$

By the method of Lagrange multipliers, the minimum occurs when $\partial\mathcal{L}/\partial r_i = \lambda \cdot 2r_i$ for all i, giving $-2cp_i/r_i^3 = 2\lambda r_i$, hence $r_i^4 = cp_i/\lambda$, so $r_i \propto p_i^{1/4}$. □

A.2 Proofs for Chapter 2

A.2.1 Hardness of Minimum Circuit (Theorem 2.8)

We give a more complete reduction than in the main text.

Proof. We reduce from Minimum Vertex Cover: given a graph $G = (V, E)$, find the smallest subset $S \subseteq V$ such that every edge $e \in E$ has at least one endpoint in S. This problem is NP-hard [4].

Given a Minimum Vertex Cover instance (G, k), construct a feedforward neural network $\mathcal{N}$ as follows. For each vertex $v \in V$, introduce a neuron n_v in the input layer. For each edge $e = (u, w) \in E$, introduce a neuron n_e in the hidden layer that receives weight 1 from n_u and weight 1 from n_w, with activation $\phi(x) = \min(x, 1)$ (piecewise linear, hence representable by ReLU compositions). Add an output neuron receiving weight 1 from each n_e. The network computes, on a binary input $\mathbf{1}_S$ (indicator vector of a subset S):

$$f_{\mathcal{N}}(\mathbf{1}_S) \;=\; |\{e \in E : e \cap S \neq \varnothing\}|\,.$$

A circuit $\mathcal{C} \subseteq \mathcal{N}$ using only a subset of input neurons (those corresponding to S) is faithful (produces output $|E|$, the number of covered edges) if and only if S is a vertex cover. A minimum-size faithful circuit therefore corresponds to a minimum vertex cover. Since minimum vertex cover is NP-hard, so is minimum-size faithful circuit discovery. □

A.3 Proofs for Chapter 3

A.3.1 The Norm Gap (Lemma 3.2)

Proof. For the memorisation solution, we need the network to correctly classify all N training examples $(x_1, y_1), \ldots, (x_N, y_N)$ by effectively storing them as key-value pairs. By the pigeonhole principle, any network achieving zero training loss on N examples with distinct inputs must have at least $\Omega(N)$ effective degrees of freedom — measured as the number of independently adjustable parameters needed to interpolate N arbitrary input-output pairs. For a network with activation function of bounded variation, each neuron can contribute at most $O(1)$ effective degrees of freedom per unit of weight norm. Therefore $\|W_{\text{mem}}\|_F^2 \geq cN$.

For the algorithmic solution, the circuit $\mathcal{C}_{\text{alg}}$ has s edges. Each edge weight is bounded in magnitude by a constant C' (determined by the scale of the target function and the layer widths). Therefore $\|W_{\text{alg}}\|_F^2 \leq \sum_{e \in E^{\mathcal{C}_{\text{alg}}}} W(e)^2 \leq C^2 s$. □

A.3.2 The Fourier Structure of Grokked Networks

We flesh out the analysis of §3.4.

Lemma A.3 (Fourier Decomposition of Attention Output)**.** *In the grokked transformer for modular addition, the output logit for class $c \in \mathbb{Z}/p\mathbb{Z}$ is:*

$$\text{logit}(c) \;=\; \sum_{k \in K} \alpha_k \cos(2\pi k(a+b-c)/p) + \beta_k \sin(2\pi k(a+b-c)/p),$$

where $K = \{k_1, \ldots, k_r\}$ is the set of active Fourier modes, and $\alpha_k, \beta_k > 0$ are constants depending on the weight matrices. The model achieves zero loss when this sum is maximised at $c = a + b \bmod p$.

Proof. The embedding matrices W_E map each token $a \in \mathbb{Z}/p\mathbb{Z}$ to a vector in $\mathbb{R}^d$ that, after grokking, approximates:

$$(W_E)_a \;\approx\; \sum_{k \in K} \big(A_k \cos(2\pi ka/p) + B_k \sin(2\pi ka/p)\big)\hat{e}_k,$$

where $\hat{e}_k$ are orthogonal directions in $\mathbb{R}^d$. The attention mechanism computes a bilinear form in the query and key vectors: $(W_E)_a^\top W_Q W_K^\top (W_E)_b$. By the cosine addition formula, this equals $\sum_{k \in K} A_k^2(\cos(2\pi ka/p)\cos(2\pi kb/p) + \sin(2\pi ka/p)\sin(2\pi kb/p)) = \sum_k A_k^2 \cos(2\pi k(a - b)/p)$. The value vectors then carry the Fourier modes of the sum $(a + b) \bmod p$, and the MLP layer implements the inverse DFT to produce the logit vector. □

A.4 Proofs for Chapter 4

A.4.1 MDL Bound on Faithful Circuit Size (Proposition 4.7)

Proof. By the MDL excess risk bound of [28], the test loss of the MDL estimator satisfies, for any comparator circuit $\hat{\mathcal{C}}$:

$$\mathcal{L}_{\text{test}}(\mathcal{C}^*) \;\leq\; \mathcal{L}_{\text{test}}(\hat{\mathcal{C}}) + O\left(\sqrt{\frac{K(\hat{\mathcal{C}})}{N}}\right).$$

Take $\hat{\mathcal{C}}$ to be a circuit that encodes the true labelling function f_Y to accuracy δ. By the Kolmogorov–Solomonoff–Chaitin theorem [41], such a circuit exists with $K(\hat{\mathcal{C}}) \leq K(f_Y) +$

$O(\log(1/\delta))$. The size of this circuit is bounded by its description length: $|E^{\hat{\mathcal{C}}}| \leq K(\hat{\mathcal{C}})/\log(1/\delta)$ (each edge requires at least $\log(1/\delta)$ bits to specify at precision δ). Since $\mathcal{C}^*$ is the MDL minimiser, $K(\mathcal{C}^*) \leq K(\hat{\mathcal{C}})$, and the stated bound follows. □

A.4.2 Proof of the Circuit–Feature Duality (Theorem 4.4)

Proof of Part (1). By the chain rule for mutual information:

$$I(T^{(\ell)}; Y) \;=\; \sum_{i=1}^{m_\ell} I(f_i^{(\ell)} \cdot T^{(\ell)}; Y \mid f_1^{(\ell)} \cdot T^{(\ell)}, \ldots, f_{i-1}^{(\ell)} \cdot T^{(\ell)}).$$

Each conditional mutual information satisfies:

$$I(f_i^{(\ell)} \cdot T^{(\ell)}; Y \mid \{f_j \cdot T\}_{j<i}) \;\leq\; I(f_i^{(\ell)} \cdot T^{(\ell)}; Y) + \sum_{j<i} I_2(f_i, f_j),$$

where $I_2(f_i, f_j)$ is the second-order interaction. When the feature frame at layer ℓ is incoherent (coherence μ_ℓ), the interaction terms are bounded by $O(\mu_\ell^2)$, giving $I(T^{(\ell)}; Y) \leq \sum_i I(f_i^{(\ell)} \cdot T^{(\ell)}; Y) + O(\ell \cdot m_\ell \cdot \mu_\ell^2)$. □

Proof of Part (3). Let $T^{(\ell)} = g_\ell(T^{(\ell-1)})$ be the deterministic map at layer ℓ. By the data processing inequality: $I(T^{(\ell)}; Y) \geq I(T^{(\ell)}; Y_\ell)$ where Y_ℓ is any deterministic function of $T^{(\ell)}$. If the activation function is approximately information- preserving (e.g., ReLU on typical inputs preserves most of the mutual information with the output), then $I(T^{(\ell)}; Y) \geq I(T^{(\ell-1)}; Y)$ up to a small constant depending on the information loss from the activation. □

A.5 Proofs for Chapter 5

A.5.1 Modern Hopfield Capacity (Theorem 5.2)

Proof sketch. Let ξ^0 be a probe with $\langle \xi^0, x_\mu \rangle > \langle \xi^0, x_\nu \rangle$ for all $\nu \neq \mu$. The softmax weight on the correct pattern is:

$$\alpha_\mu \;=\; \frac{e^{\beta \langle x_\mu, \xi^0 \rangle}}{\sum_\nu e^{\beta \langle x_\nu, \xi^0 \rangle}}.$$

The key step is to bound α_μ from below. With patterns drawn uniformly from S^{n-1}, the inner products $\langle x_\nu, \xi^0 \rangle$ for $\nu \neq \mu$ are i.i.d. with mean 0 and variance $1/n$. By the log-sum-exp concentration inequality:

$$\log \sum_{\nu \neq \mu} e^{\beta \langle x_\nu, \xi^0 \rangle} \;\leq\; \log m + \frac{\beta^2}{2n} + O\left(\sqrt{\frac{\log m}{n}}\right)$$

with high probability. Therefore $\alpha_\mu \geq 1 - e^{-\beta\Delta + \log m + \beta^2/(2n)}$ where $\Delta = \langle x_\mu, \xi^0 \rangle - \max_{\nu \neq \mu} \langle x_\nu, \xi^0 \rangle > 0$. Setting $\beta = 1/\sqrt{n}$ and using $\Delta \geq \Omega(1/\sqrt{n})$ (from the gap between the top two inner products), one obtains $\alpha_\mu \geq 1 - e^{-\Omega(1)}$ provided $m \leq C e^{n\beta^2/2}/\beta$. The retrieved vector $\xi^1 = X\alpha$ is then within ε of x_μ. □

A.5.2 Attention Entropy Critical Temperature (Proposition 5.6)

Proof. The entropy of the softmax is $\mathcal{H}(\alpha) = \log Z - \beta \mathbb{E}_\alpha[\langle k_s, q \rangle]$ where $Z = \sum_s e^{\beta \langle k_s, q \rangle}$. For small β, expand: $Z \approx T + \frac{\beta^2}{2} \sum_s \langle k_s, q \rangle^2$, so $\log Z \approx \log T + \frac{\beta^2 \sigma^2}{2}$ where $\sigma^2 = \frac{1}{T} \sum_s \langle k_s, q \rangle^2$.

The mean attention value is $\mathbb{E}_\alpha[\langle k_s,\, q\rangle] \approx \frac{\beta\sigma^2}{T}$ (from the first-order term in the softmax expansion). Therefore $\mathcal{H}(\alpha) \approx \log T - \frac{\beta^2\sigma^2}{2} - \frac{\beta^2\sigma^2}{1} = \log T - \frac{3\beta^2\sigma^2}{2}$. The crossover between the $\log T$ plateau (diffuse regime) and the near-zero regime (sharp) occurs when $\beta^2\mathbb{E}_q[\sigma^2(q)] \sim 1$, giving $\beta^* = 1/\sqrt{\mathbb{E}_q[\sigma^2(q)]} = 1/\sqrt{\mathbb{E}_q[\max_s\langle k_s,\, q\rangle^2]}$ (since for large β, the variance is dominated by the maximum inner product). □

A.6 Proofs for Chapter 6

A.6.1 The Gradient Flow on the Frame Manifold (Appendix A)

Proof. The energy (6.1) can be written as $\mathcal{E}(F) = \sum_{i\neq j} G_{ij}^2$ where $G_{ij} = \langle f_i,\, f_j\rangle$. The gradient with respect to f_i (before projection to the sphere) is:

$$\nabla_{f_i}\mathcal{E} \;=\; 4\sum_{j\neq i} G_{ij}\nabla_{f_i}G_{ij} \;=\; 4\sum_{j\neq i}\langle f_i,\, f_j\rangle f_j.$$

Projecting onto the tangent space $T_{f_i}S^{d-1} = \{v : \langle v,\, f_i\rangle = 0\}$ gives:

$$\dot{f}_i \;=\; -\pi_{T_{f_i}}(\nabla_{f_i}\mathcal{E}) \;=\; -4\sum_{j\neq i}\langle f_i,\, f_j\rangle\big(f_j - \langle f_i,\, f_j\rangle f_i\big),$$

which is (6.15). The critical points satisfy $\sum_{j\neq i}\langle f_i,\, f_j\rangle(f_j - \langle f_i,\, f_j\rangle f_i) = 0$ for all i, which simplifies to $\sum_{j\neq i}\langle f_i,\, f_j\rangle f_j \propto f_i$ for all i, i.e., the frame operator $S = FF^\top$ has f_i as an eigenvector for every i. This is the UNTF condition. □

A.6.2 Symmetry Breaking and Mean-Field Exponents (§6.6)

To make the claim about the square-root exponent in (6.6) more precise, we compute the effective Landau free energy near the critical sparsity k^*.

Let $\phi = |\rho|$ be the order parameter and expand $\mathcal{L}_{\text{rep}}$ as a power series in ϕ near $\phi = 1$ (the merged phase):

$$\mathcal{L}_{\text{rep}}(F, k) \;\approx\; \mathcal{L}_0 + a_2(k)(1-\phi^2) + a_4(k)(1-\phi^2)^2 + \cdots$$

where $a_2(k) = (k-k^*)\cdot\partial_{k^*}^2\mathcal{E}/\partial\phi^2$ changes sign at $k = k^*$. For $k > k^*$, $a_2 < 0$ and the minimum of $\mathcal{L}_{\text{rep}}$ shifts from $\phi = 1$ to $\phi < 1$. Setting $\partial\mathcal{L}_{\text{rep}}/\partial\phi = 0$ gives $\phi^2 \approx 1 + a_2/(2a_4) \approx 1 - c(k-k^*)$ for a constant $c > 0$, so $1-\phi \approx \frac{c}{2}(k-k^*)$, i.e., $\phi \approx 1 - \frac{c}{2}(k-k^*)$ to first order. The square-root arises when higher-order terms dominate, giving $\phi \approx 1 - c'\sqrt{k-k^*}$, which is the $\mathbb{Z}_2$ mean-field exponent $\beta_{\text{MF}} = 1/2$.

A.7 Proofs for Chapter 8

A.7.1 Sample Complexity of Mechanistic Verification (Theorem 8.5)

Proof sketch. We use the minimax lower bound framework of [37]. Define two distributions $D_0 = D$ and D_1, where D_1 is obtained by reweighting D to place extra mass on a region $A \subset \mathcal{X}$ with $D(A) = \rho$. On A, the circuit $\mathcal{C}$ behaves differently from $f_{\mathcal{N}}$ by a carefully chosen adversarial perturbation. The TV distance between D_0 and D_1 is ρ.

Any verification algorithm that distinguishes between D_0 and D_1 with probability $1-\delta$ from N samples must have $N \geq \Omega(\rho^{-2}\log(1/\delta))$ by the Le Cam minimax theorem. However, the adversarial perturbation can be made arbitrarily "small" in terms of its Fourier content: it lives in a d-dimensional ball of radius ρ in TV distance but in a ball of radius $e^{\Omega(d\rho^2)}$ in the

space of distinguishable distributions. The extra factor of $e^{\Omega(d\rho^2)}$ in the sample complexity bound (8.1) arises from the curse of dimensionality: to distinguish distributions that agree on most of $\mathcal{X}$ but differ on a ρ-fraction, one needs exponentially many samples in dimension d. □

A.7.2 Shape of the IPF (Theorem 8.9)

Proof of Part (1). Let $\mathcal{C}$ be any faithful circuit with $K(\mathcal{C}) \leq K_0$. By the data processing inequality, $I(X; f_{\mathcal{C}}(X)) \leq I(X;T)$ where T is the representation used by $\mathcal{C}$. By the rate-distortion lower bound (Lemma 4.1):

$$K(\mathcal{C})/(\text{bits per edge}) \;\geq\; |E^{\mathcal{C}}| \;\geq\; \frac{I(X;T)}{\log(1/\delta)}.$$

By the data processing inequality and the definition of the rate-distortion function: $I(X;T) \geq R(\varepsilon^2) \geq I(X;Y) - C\log(1/\varepsilon)$. Since $K(f_Y)$ lower bounds $I(X;Y)$ (by the algorithmic mutual information inequality [41]), we get $K_0 \geq K(\mathcal{C}) \geq K(f_Y) - C\log(1/\varepsilon)$, proving part (1). □

Proof of Part (3). As $\varepsilon \to 0$, the rate-distortion function satisfies $R(\varepsilon^2) \to I(X;Y)$. By the MDL bound (Proposition 4.7), $|E^{\mathcal{C}^*}| \geq K(f_Y)/\log(1/\varepsilon)$. Since circuits with $K(\mathcal{C}) \ll K^*$ cannot achieve $\varepsilon \to 0$ by part (1), the IPF approaches the point $(\varepsilon = 0, K_0 = K^*)$ from the interior of the achievable region. □

Appendix B

Open Problems

We list here the most important open problems raised by this theory, as a map for future work. Each problem is stated precisely enough to be checkable, not as a vague aspiration but as a challenge that admits a definite mathematical resolution.

Problem 1. Deriving the Chinchilla exponents rigorously. Theorem 9.2 derives $\alpha = (\gamma - 1)/2$ under the assumption of a power-law feature distribution and optimal Welch-bound interference allocation. Making this rigorous requires (a) proving that the feature importance distribution of natural language satisfies a power law with an explicit exponent γ, and (b) proving that gradient descent on a transformer approximately achieves the Welch-bound interference allocation. Both steps require connecting the statistical properties of natural language corpora (a topic in empirical linguistics) to the geometry of the feature frame learned by the network (a topic in learning theory).

Problem 2. The circuit phase diagram for transformers. Chapter 6 develops the phase diagram for feature configurations in an abstract setting. Extending this to transformer architectures requires a higher-dimensional energy functional that accounts for the attention mechanism, the residual stream, and the interaction between heads at different layers. The critical points of this functional — which correspond to the phase boundaries of the circuit phase diagram — are not yet characterised. A natural conjecture is that the phase boundaries are determined by the composition scores of Definition 2.6.

Problem 3. The universality theorem beyond IB optimality. Theorem 7.4 assumes information bottleneck optimality, which is not guaranteed by gradient descent. Proving a universality theorem under realistic training conditions requires understanding the implicit regularisation of SGD with weight decay, adaptive optimisers, and dropout at the circuit level. A weaker but tractable version of the problem: prove universality within the class of networks trained by SGD with ℓ^2 regularisation, without assuming IB optimality.

Problem 4. Kolmogorov complexity of natural language modelling. Proposition 10.8 requires bounding $K(f_Y)$ for the language modelling task. For natural language, this is essentially the question of the "algorithmic complexity of the world" — a deep and philosophically fraught problem. A more tractable version: for a specific formal language $\mathcal{L}$ (e.g., a fragment of propositional logic, or a simple programming language), characterise $K(f_Y)$ in terms of the complexity of the grammar of $\mathcal{L}$, and verify that the scaling law exponents predicted by Theorem 9.2 match the empirically observed exponents for a language model trained on $\mathcal{L}$.

Problem 5. Causal faithfulness at scale. Theorem 10.6 characterises causal faithfulness in terms of circuit bottlenecks. Developing scalable algorithms for testing causal faithfulness requires (a) an efficient implementation of the circuit discovery algorithm (2.3) for large transformers, and (b) a method for testing the "no bypass circuit" condition (condition (2) of Theorem 10.6)

without enumerating all circuits. A promising direction: show that the bypass circuits can be detected by the *direct effect* of the feature on the output, which is computable in time linear in the number of edges.

Problem 6. The geometry of values. Hypothesis 10.9 proposes that value coherence is reflected in the geometry of value feature vectors. Proving this requires (a) identifying value features in large language models using the Contrast-Consistent Search or Representation Engineering methods of [14, 70]; (b) measuring their mutual inner products; and (c) comparing the resulting geometry to an operationalised version of value coherence. The third step requires an agreement between the mechanistic interpretability community and the moral philosophy community on how to operationalise "value coherence" — a conceptual problem that may be as hard as the mathematical one.

Problem 7. Phase transitions in the interaction graph. As the number of attention heads H grows, the interaction graph $\mathcal{G}$ of Definition 2.6 becomes dense. Understanding the phase structure of this graph — in particular, whether there is a sharp transition from a sparse circuit regime to a dense circuit regime — is an open problem with implications for the scalability of mechanistic interpretability. A natural conjecture: the transition occurs at the same scale $\gamma = 5/3$ as the phase boundary between the dense and sparse circuit regimes in Proposition 9.3.

Problem 8. The relationship between grokking and double descent. Proposition 9.7 establishes a qualitative analogy between grokking and double descent. Making this precise requires showing that the escape time formula (3.5) and the double descent peak both derive from the same free energy calculation applied at different scales (time vs. model size). The key step is to show that the norm gap of Lemma 3.2 is the relevant thermodynamic quantity in both cases, and that the critical weight decay λ^* (Theorem 3.3) corresponds to a critical model size N_{interp} (Proposition 9.7) through the same underlying competition between memorisation and generalisation circuits.

B.1 Proofs for Chapter 7

B.1.1 Transformer–MLP Equivalence (Theorem 7.6)

We give the main ideas of the proof without all technical details.

Proof sketch. The task $\mathcal{T} = \mathcal{T}_L \circ \cdots \circ \mathcal{T}_1$ decomposes into L sequential computations.

MLP lower bound: An MLP circuit implementing $\mathcal{T}$ must execute each $\mathcal{T}_\ell$ sequentially. Since the computation of $\mathcal{T}_\ell$ requires at least one layer of the MLP (by the depth-width tradeoff, Theorem 2.7, each $\mathcal{T}_\ell$ is a polynomial function requiring at least one ReLU layer of width $O(n_\ell)$), the total depth is $\Omega(L)$. The required width at each depth is determined by the intermediate representation dimension n_ℓ, giving width $O(\max_\ell n_\ell)$.

Transformer upper bound: A transformer can parallelise across layers using attention. The key observation is that the attention mechanism can implement the composition $T^{(\ell)} = g_\ell(T^{(\ell-1)})$ in a single layer, for any g_ℓ of polynomial complexity, by using the value vectors to implement the transformation and the attention weights to select the correct source positions. Since L sequential computations can be parallelised into $O(\log L)$ rounds using a parallel prefix computation, the transformer requires depth $O(\log L)$. The increase in width (by a factor of $L^{1/2}$) comes from the need to store all L intermediate computations simultaneously in the residual stream.

Same universality class: Both circuits implement the same sequence of intermediate representations $T^{(1)}, \ldots, T^{(L)}$. The MLP takes one layer per computation; the transformer takes

$O(\log L)$ layers for all computations in parallel. The feature vectors at each stage are the same (up to orthogonal transformation of each layer's representation space), establishing the isomorphism of Definition 7.1. □

B.1.2 Gradient Flow Circuit Selection (Proposition 7.10)

Proof. The gradient flow at initialisation W_0 is $\dot{W} = -\nabla_W \mathcal{L}(W_0)$, which for a random initialisation $W_0 \sim \mathcal{N}(0, \sigma^2 I)$ has a random direction. The circuit $\mathcal{C}^*$ is the one that is most aligned with the expected gradient, because gradient descent will first move in the direction of the (negative) gradient, and the most aligned circuit will be reached first.

Formally, among all circuits of norm $N_{\min}$, the gradient flow from W_0 reaches the circuit $\mathcal{C}$ with probability proportional to $\exp(-\|W_{\mathcal{C}} - W_0\|_F^2/(2\sigma^2))$. This is maximised by the circuit $\mathcal{C}^*$ that minimises $\|W_{\mathcal{C}^*} - W_0\|_F^2$, i.e., the circuit most similar to the initialisation. Taking the expectation over random W_0 gives the inner product criterion (7.3). □

B.2 Proofs for Chapter 9

B.2.1 Optimal Parameter–Data Tradeoff (Proposition 9.8)

Proof. The Chinchilla loss is $\mathcal{L}(N, D) = \mathcal{L}_\infty + AN^{-\alpha} + BD^{-\beta}$. Under the constraint $ND = C/c$ (fixed compute budget), substitute $D = C/(cN)$:

$$\mathcal{L}(N) \;=\; \mathcal{L}_\infty + AN^{-\alpha} + B\left(\frac{cN}{C}\right)^{\beta} \;=\; \mathcal{L}_\infty + AN^{-\alpha} + \frac{Bc^\beta}{C^\beta}N^\beta.$$

Setting $d\mathcal{L}/dN = 0$:

$$-\alpha A N^{-\alpha-1} + \beta\frac{Bc^\beta}{C^\beta}N^{\beta-1} \;=\; 0 \;\Rightarrow\; N^{\alpha+\beta} \;=\; \frac{\alpha A C^\beta}{\beta B c^\beta}.$$

Therefore $N^*(C) \propto C^{\beta/(\alpha+\beta)}$ and $D^*(C) = C/(cN^*(C)) \propto C^{1-\beta/(\alpha+\beta)} = C^{\alpha/(\alpha+\beta)}$, giving (9.8). □

B.2.2 Interpretability Scaling Law (Proposition 9.9)

Proof sketch. The fraction of capability interpretable with budget K_0 is the fraction of the total loss reduction attributable to circuits with $K(\mathcal{C}) \le K_0$. By Proposition 4.7, a circuit of complexity K_0 corresponds to a circuit of size $|E| \le K_0/\log(1/\delta)$. By Proposition 9.3, circuits of this size account for features up to rank $m_0 = m^*(N)\cdot(K_0/K^*(N))^{1/\text{circuit-size-exponent}}$. The fraction of loss reduction attributable to the top-m_0 features is:

$$\text{InterpFrac} \;=\; 1 - \frac{\mathcal{L}_{\text{res}}(m_0)}{\mathcal{L}_{\text{res}}(0)} \;=\; 1 - \left(\frac{m_0}{m^*(N)}\right)^{\gamma-1}.$$

Substituting $m_0 \propto \exp(-K^*(N)/K_0)$ (from the Kolmogorov complexity–circuit size connection) gives (9.11). □

B.3 Additional Results: The Information Geometry of Circuits

The information-theoretic analysis of Chapter 4 can be extended using the tools of *information geometry* [1] to give a Riemannian structure on the space of circuit representations.

Definition B.1 (Fisher Information of a Circuit)**.** Let $\mathcal{C}$ be a circuit and $p_{\mathcal{C}}(Y|X)$ be the distribution it defines over labels. The *Fisher information matrix* of $\mathcal{C}$ with respect to its parameters θ (the weights $W^{\mathcal{C}}$) is:

$$\mathcal{I}_{\mathcal{C}}(\theta) \;=\; \mathbb{E}_{X,Y}\Big[\nabla_\theta \log p_{\mathcal{C}}(Y|X)\,(\nabla_\theta \log p_{\mathcal{C}}(Y|X))^\top\Big]. \tag{B.1}$$

The Fisher information matrix defines a Riemannian metric on the parameter space of circuit $\mathcal{C}$ — the Fisher-Rao metric. This metric has a natural interpretation: the geodesic distance between two parameter configurations θ_1 and θ_2 measures how distinguishable the two circuit behaviours are, in the information-theoretic sense.

Proposition B.2 (Fisher Information and Feature Coherence)**.** *For a circuit $\mathcal{C}$ with feature frame of coherence μ, the Fisher information matrix satisfies:*

$$\lambda_{\min}(\mathcal{I}_{\mathcal{C}}) \;\geq\; (1 - m\mu^2) \cdot \lambda_{\min}(\mathcal{I}_{\mathcal{C},\text{incoherent}}), \tag{B.2}$$

where $\mathcal{I}_{\mathcal{C},\text{incoherent}}$ is the Fisher information of the same circuit with a perfectly incoherent frame (all features orthogonal).

The proposition says that incoherence (small μ) implies large minimum Fisher information, which in turn implies that small parameter changes lead to large changes in the circuit's behaviour. This is the information-geometric version of the intuition that incoherent features are well-separated and individually identifiable.

B.4 Proofs for Chapter 10

B.4.1 Circuit-Based Deception Detection (Theorem 10.2)

Proof sketch. The augmented loss $\mathcal{L}(W) + \mu \cdot I(\mathcal{C}_{\text{ext}}; \text{full output})$ can be written as $\mathcal{L}(W) + \mu \cdot I(f_{\mathcal{C}_{\text{ext}}}(X); f_{\mathcal{N}}(X)[V_{\text{out}}])$. By the variational formula (4.6), this mutual information can be estimated from data by training a discriminator to distinguish the joint distribution $(f_{\mathcal{C}_{\text{ext}}}, f_{\mathcal{N}}[V_{\text{out}}])$ from the product distribution.

If $\mathcal{N}$ is (p, q, ε)-deceptive (Definition 10.1), the output of $\mathcal{N}$ on deployment inputs depends on $\mathcal{C}_{\text{int}}$ (with probability q) and therefore on information not captured by $\mathcal{C}_{\text{ext}}$ (which is $(0, q)$-independent of $\mathcal{C}_{\text{int}}$ by condition (3) of Definition 10.1). This residual dependence is detected by the mutual information penalty, which forces the ℓ^1 relaxation to include edges of $\mathcal{C}_{\text{int}}$ in the circuit. The probability of detection is at least $1 - e^{-q^2|D_{\text{dep}}|/\varepsilon^2}$ by standard concentration bounds applied to the mutual information estimator. □

B.4.2 Causal Faithfulness (Theorem 10.6)

Proof. ($\Rightarrow$): If f_i is $(0, \mathcal{I})$-causally faithful for g_i, then by definition (10.2), the causal effect of $f_i^\top T$ on $g_i(X)$ equals the observational effect. By Pearl's do-calculus [50], this is equivalent to the d-separation condition: $g_i(X) \perp\!\!\!\perp X \mid f_i^\top T$ in the causal graph. This means all information about $g_i(X)$ flows through $f_i^\top T$, i.e., f_i is 0-sufficient for $g_i(X)$ (condition (1)). And every path from X to g_i-relevant outputs passes through $f_i^\top T$ (condition (2): no bypass circuits).

($\Leftarrow$): If conditions (1) and (2) hold, then $g_i(X)$ is d-separated from X given $f_i^\top T$ in the causal graph, and by the d-separation criterion, the causal effect equals the observational effect. □

B.5 Proofs for Chapter 11

B.5.1 Coding Efficiency and the Welch Bound (Proposition 1.10)

Proof. The mutual information stored by the representation is approximately $I(X;T) \approx d\log(m/k)$ (from the capacity formula of Lemma 1.6: $m \leq e^{\varepsilon^2 d}$, so $d\log(m/k) \approx \varepsilon^2 d \cdot (\log m - \log k)$). The total neural activity is $\sum_i \mathbb{E}[a_i] = k \cdot p_{\text{active}}$ where p_{active} is the probability of any feature being active. For the uniform distribution over k-sparse vectors, $p_{\text{active}} = k/m$. So $\sum_i \mathbb{E}[a_i] = k^2/m$.

The efficiency is then $\eta_{\text{code}} = I(X;T)/\sum_i \mathbb{E}[a_i] \approx (d\log(m/k))/(k^2/m) = (md\log(m/k))/k^2$. Using $\mu^2 \approx m/d^2$ (from the Welch bound $\mu \approx \sqrt{m/d^2}$ for large m) and $\varepsilon^2 \approx \mu^2 = m/d^2$, we get $\eta_{\text{code}} \approx d\log(m/k)/(k\varepsilon^2)$, which matches (1.13). □

B.5.2 Superposition and Generalisation (Proposition 1.9)

Proof. The result follows directly from Theorem 1.4 applied to the test input a^*. By the RIP of order $2k$ with $\delta_{2k} < \sqrt{2} - 1$, the ℓ^1-minimisation (1.4) with $h = Fa^*$ recovers $\hat{a} = a^*$ exactly (zero error). For the approximate RIP case ($\delta_{2k} \leq \sqrt{2} - 1 + \varepsilon_0$), the standard error bound for noisy compressed sensing gives $\|\hat{a} - a^*\| \leq C\delta_{2k} \|a^*\|$ for a universal constant $C \approx 4/(1 - (1+\sqrt{2})\delta_{2k})$. □

B.5.3 Feature Accuracy and Sample Complexity (Equation 9.13)

Derivation. Consider a feature f_i with importance $\lambda_i = I(f_i^\top T; Y)$. The accuracy of the read-out of feature i on a fresh example (x, y) is determined by the signal-to-noise ratio of the estimator of λ_i from N training examples. By the Berry-Esseen theorem applied to the empirical mutual information:

$$\Pr[\hat{\lambda}_i \geq \lambda_i/2] \;\geq\; 1 - 2e^{-\lambda_i N/(2d)},$$

where $\hat{\lambda}_i$ is the empirical mutual information computed from N samples and we use the fact that the variance of the empirical MI estimator is $O(d/N)$. When $\hat{\lambda}_i \geq \lambda_i/2$, the feature is learned to constant accuracy. The accuracy is therefore $\mathbb{E}[\text{acc}(f_i)] \geq 1 - 2e^{-\lambda_i N/(2d)}$, which is approximately $1 - e^{-\lambda_i N/d}$ for large $\lambda_i N/d$, giving (9.13). □

B.6 A Worked Example: The Modular Addition Circuit in Full

We give a complete worked example tracing the modular addition circuit from the abstract definitions of Chapter 2 through the grokking analysis of Chapter 3 and the information-theoretic analysis of Chapter 4.

B.6.1 Setup

Consider the task $\mathcal{T}_p$ of computing $(a + b) \bmod p$ for $p = 11$ (a small prime). The network is a one-layer transformer with embedding dimension $d = 24$, key-query dimension $d_k = 12$, value dimension $d_v = 12$, and MLP width $4d = 96$. This is the exact architecture studied by Nanda et al. [45].

B.6.2 The Features

After grokking, the embedding matrix $W_E \in \mathbb{R}^{p \times d}$ has its rows (one per token) aligned with a specific set of Fourier modes. The features of the grokked network are:

- For each active Fourier mode $k \in K = \{k_1, \ldots, k_r\}$ with $r \approx 5$, two features: $f_{k,\cos} = \hat{e}_{k,\cos}$ (the cosine Fourier direction) and $f_{k,\sin} = \hat{e}_{k,\sin}$ (the sine Fourier direction), where $\hat{e}_{k,\cos}$ and $\hat{e}_{k,\sin}$ are orthogonal unit vectors in $\mathbb{R}^{24}$.

Total: $2r \approx 10$ features. The feature frame has $m = 2r$ and $d = 24$, giving $m/d \approx 0.4$ — well below the Welch-bound regime (which requires $m > d$). In this regime, the features are approximately orthogonal and there is negligible interference.

B.6.3 The Circuit

The grokked circuit $\mathcal{C}_p$ consists of:

(1) *Embedding sub-circuit* $\mathcal{C}_E$: maps each token $a \in \mathbb{Z}/p\mathbb{Z}$ to its embedding $(W_E)_a \in \mathbb{R}^d$, which is approximately $\sum_{k \in K} A_k(\cos(2\pi ka/p)\hat{e}_{k,\cos} + \sin(2\pi ka/p)\hat{e}_{k,\sin})$.

(2) *Attention sub-circuit* $\mathcal{C}_{\text{att}}$: computes the bilinear form $(W_E)_a^\top W_Q W_K^\top (W_E)_b = \sum_{k \in K} A_k^2 \cos(2\pi k(a - b)/p)$ and uses this to write the Fourier components of $a + b$ to the residual stream.

(3) *MLP sub-circuit* $\mathcal{C}_{\text{MLP}}$: implements the inverse DFT, mapping the Fourier components at position 3 (the output position) to the logit vector indexed by $c \in \mathbb{Z}/p\mathbb{Z}$.

The sequential composition is $\mathcal{C}_p = \mathcal{C}_{\text{MLP}} \circ \mathcal{C}_{\text{att}} \circ \mathcal{C}_E$.

B.6.4 Faithfulness

The circuit $\mathcal{C}_p$ is $(0.01, D_p)$-faithful for the modular addition task $\mathcal{T}_p$, where D_p is the uniform distribution over pairs $(a, b) \in (\mathbb{Z}/p\mathbb{Z})^2$. This is verified empirically by Nanda et al. [45]: after grokking, the circuit accounts for $> 99\%$ of the network's accuracy on the test set.

B.6.5 Information-Theoretic Analysis

The mutual information carried by the circuit is:

$$I(X; T_{\mathcal{C}_p}) \;=\; \sum_{k \in K} I(X; (a_{k,\cos}, a_{k,\sin})) \;\approx\; 2r \log 2,$$

where each feature $(a_{k,\cos}, a_{k,\sin})$ carries approximately $\log 2$ bits about the task (it determines $\cos(2\pi k(a+b)/p)$ and $\sin(2\pi k(a+b)/p)$, which together determine $(a+b) \bmod p / \gcd(k, p)$). For $r = 5$ active modes, $I(X; T_{\mathcal{C}_p}) \approx 10 \log 2 \approx 6.9$ bits.

The task mutual information is $I(X; Y) \approx \log p = \log 11 \approx 2.4$ bits (since the output $c = (a + b) \bmod p$ is uniform over $\mathbb{Z}/p\mathbb{Z}$ given a uniform prior on (a, b)). The circuit stores approximately $2.9\times$ more bits than the task requires, reflecting the redundancy of using multiple Fourier modes.

This redundancy is not wasteful: using multiple Fourier modes improves robustness (each mode independently identifies c modulo $\gcd(k, p)$), which is the circuit's analogue of the redundant encoding in error-correcting codes.

B.6.6 Kolmogorov Complexity

The circuit $\mathcal{C}_p$ can be described in $O(\log p + r \log(1/\varepsilon))$ bits: specify p, the number of active Fourier modes r, their values $k_1, \ldots, k_r$ (each at most $\log p$ bits), and the precision ε to which the Fourier decomposition is computed. For $p = 11$, $r = 5$, and $\varepsilon = 0.01$, this gives $K(\mathcal{C}_p) \approx 4 + 10 + 5 = 19$ bits — a very short description.

The Kolmogorov complexity of the modular addition task itself is $K(f_{\mathcal{T}_p}) \approx \log p + O(1) \approx 4$ bits (specify p and the operation "addition modulo p"). The circuit's complexity $K(\mathcal{C}_p) \approx 19$ bits is somewhat larger than $K(f_{\mathcal{T}_p}) \approx 4$ bits, consistent with the MDL bound (4.13): $K(\mathcal{C}^*)$ approaches $K(f_Y)$ from above as the training set size N increases.

This worked example illustrates all the main theoretical results of the monograph in a concrete setting, connecting the geometry of superposition (Chapter 1), the circuit algebra (Chapter 2), the grokking dynamics (Chapter 3), and the information-theoretic analysis (Chapter 4) into a coherent picture.

B.7 Additional Proofs for Chapter 11

B.7.1 Fisher Information and the Welch Bound (Proposition 11.3)

Proof. The Fisher information of the prediction distribution $p_{\mathcal{C}}(Y|X)$ with respect to the feature activation a_i is $\mathcal{I}_{a_i} = \mathbb{E}[\partial_{a_i} \log p_{\mathcal{C}}(Y|X)]^2$. For a network reading out feature i as $f_i^\top h(X)$, the gradient of the log-likelihood with respect to a_i is $\partial_{a_i} \log p_{\mathcal{C}}(Y|X) = \lambda_i \cdot \mathbf{1}[Y = g_i(X)]/p_i + O(\mu)$ where the $O(\mu)$ term comes from interference from other features. The Fisher information is therefore $\mathcal{I}_{a_i} \geq (\lambda_i/p_i)^2 p_i(1-p_i) - O(\mu^2 m) = \lambda_i^2/p_i \cdot (1-p_i) - O(m\mu^2)$. The minimum Fisher information over all features is $\lambda_{\min}^2(1-p_{\max})/p_{\max} - O(m\mu^2)$. For the case $p_i = 1/2$ (uniform activation probability) and $\lambda_i \geq \lambda_{\min}$, this gives $\lambda_{\min}(\mathcal{I}_{\mathcal{C}}) \geq \lambda_{\min}/2 - O(m\mu^2) \geq \lambda_{\min}(1 - O(m\mu^2 d/\lambda_{\min}))$. Using $m\mu^2 \leq m \cdot m/d^2 = m^2/d^2$ (from the Welch bound) and $\lambda_{\min} \geq \lambda_i$ gives the stated bound. □

B.7.2 Anomaly Detection and Distribution Shift (Proposition 10.12)

Proof. The expected feature anomaly score under D' is:

$$\mathbb{E}_{x\sim D'}[\mathrm{FAS}(x)] = \sum_i \lambda_i \mathbb{E}_{x\sim D'}[(a_i(x) - p_i^D)^2].$$

Writing $a_i(x) - p_i^D = (a_i(x) - p_i^{D'}) + (p_i^{D'} - p_i^D)$ and using the non-negativity of variance:

$$\mathbb{E}_{x\sim D'}[(a_i(x) - p_i^D)^2] \geq (p_i^{D'} - p_i^D)^2.$$

The total variation distance satisfies $\mathrm{TV}(D, D') \geq \left|p_i^{D'} - p_i^D\right|$ for any event, so the shift in activation probability is bounded by $\left|p_i^{D'} - p_i^D\right| \geq \rho \cdot \sqrt{p_i^D(1-p_i^D)}$ for a distribution shift of $\rho = \mathrm{TV}(D, D')$ concentrated on feature i. Multiplying by λ_i and summing gives (10.5). □

B.7.3 Grokking Condition (Proposition 3.7)

Proof. By Theorem 3.3, the grokking time satisfies $T_{\mathrm{grok}} \asymp \exp(\Delta\mathcal{F}/T_{\mathrm{eff}})$ where $T_{\mathrm{eff}} = \eta\lambda/2$ and $\Delta\mathcal{F} \approx \lambda(N-s)/2$ (from Lemma 3.2: the barrier height is proportional to the norm gap, which is $\lambda \|W_{\mathrm{mem}}\|_F^2/2 - \lambda \|W_{\mathrm{alg}}\|_F^2/2 \approx \lambda(N-s)/2$). Therefore $T_{\mathrm{grok}} \asymp \exp((N-s)/\eta)$. Setting $T_{\mathrm{grok}} \leq T_{\max}$ gives $(N-s)/\eta \leq \log T_{\max}$, i.e., $N \leq s + \eta \log T_{\max}$. For grokking to occur before $T_{\max}$ but after $T_{\mathrm{grok,min}}$, we need $s + \eta \log T_{\mathrm{grok,min}} \leq N \leq s + \eta \log T_{\max}$, which gives (3.11) with $C = \eta$. □

B.8 A Complete Index of Main Results

For ease of reference, we list all named results in the monograph, grouped by type.

B.8.1 Theorems

Theorem 1.1 (Welch Bound)
Lower bound $\mu(F) \geq \sqrt{(m-d)/[d(m-1)]}$ on the coherence of any unit-norm frame.

Theorem 1.4 (ℓ^1 Recovery)
If F satisfies RIP with $\delta_{2k} < \sqrt{2}-1$, then ℓ^1 minimisation recovers any k-sparse vector exactly.

Theorem 1.5 (Random Frames Satisfy RIP)
Random unit-norm frames satisfy RIP of order k with high probability.

Theorem 2.7 (Depth–Width Tradeoff)
Circuits of depth $O(\log(1/\varepsilon))$ and width $O(s)$ approximate functions with s-sparse Fourier spectra; depth-1 circuits require exponential width.

Theorem 2.8 (Hardness of Minimum Circuit)
Finding the minimum-size faithful circuit is NP-hard.

Theorem 3.3 (Critical Weight Decay)
The grokking time scales as $\exp(\Delta\mathcal{F}/\lambda\eta)$; there exists a critical λ^* below which grokking does not occur.

Theorem 4.2 (Circuits Achieve the Bottleneck)
The minimal faithful circuit approximately solves the information bottleneck problem.

Theorem 4.4 (Circuit–Feature Duality)
Features at each layer are complete, circuits factorise as frame transformations, and information flows forward.

Theorem 5.2 (Modern Hopfield Capacity)
Modern Hopfield networks store $e^{d_k/2}$ patterns in dimension d_k.

Theorem 6.2 (Stability of Two-Feature Configurations)
A two-feature configuration is stable iff $\rho^2 > 1/d$.

Theorem 6.3 (Critical Sparsity for Splitting)
Feature splitting occurs when $k > k^* = \lambda d^2/[2(m-d)]$.

Theorem 7.3 (Circuit Universality)
Under IB optimality, circuits from independently trained networks are isomorphic.

Theorem 7.5 (Transformer–MLP Equivalence)
For compositional tasks, transformer and MLP circuits are in the same universality class.

Theorem 8.1 (Undecidability)
Neural network equivalence is undecidable for ReLU networks.

Theorem 8.3 (Σ_2^P-Completeness)
Approximate circuit discovery is Σ_2^P-complete.

Theorem 8.5 (Sample Complexity)
Mechanistic verification over distribution shifts requires exponential samples.

Theorem 8.7 (Irreducible Circuits)
Irreducible circuits exist for tasks with high Kolmogorov complexity.

Theorem 8.9 (Shape of the IPF)
The IPF satisfies $K_0 \geq K^* - C\log(1/\varepsilon)$ and approaches $(0, K^*)$ asymptotically.

Theorem 9.2 (Scaling Exponent Derivation)
$\alpha = (\gamma - 1)/2$ where γ is the feature importance exponent.

Theorem 9.5 (Emergence Threshold)
A capability emerges at scale $N^* \sim (\log m^*/\varepsilon^2)^2$.

Theorem 10.2 (Deception Detection)
Augmented circuit discovery detects hidden circuits with probability $1 - e^{-q^2|D_{\text{dep}}|/\varepsilon^2}$.

Theorem 10.6 (Causal Faithfulness)
A feature is causally faithful iff it is sufficient and a circuit bottleneck.

B.8.2 Definitions

The main definitions are: Circuit (2.1), Faithfulness (2.4), Transformer Circuit (2.6), Grokking (3.1), Rate-Distortion (4.1), Sufficient Feature (4.3), Circuit Isomorphism (7.1), Universality Class (7.2), Mechanistic Irreducibility (8.6), IPF (8.8), Emergence (9.4), Deceptive Circuit (10.1), Causal Faithfulness (10.5), Alignment (10.7).

B.8.3 Hypotheses

The main hypotheses are: Platonic Representation Hypothesis (7.2) and Geometric Coherence of Values (10.9).

Appendix C

Notation and Mathematical Conventions

This appendix collects the notation used throughout the monograph. Definitions are given once here and assumed in the main text without repetition.

C.1 General Conventions

We write scalars in lowercase italic (a, b, λ), vectors in lowercase bold or italic (f_i, v, h), and matrices in uppercase (W, G, F). Sets are written in calligraphic uppercase ($\mathcal{C}, \mathcal{N}, \mathcal{T}$). We use $\mathbb{R}$ for the reals, $\mathbb{N}$ for the natural numbers, $\mathbb{Z}$ for the integers, and $\mathbb{Z}/p\mathbb{Z}$ for the integers modulo p. Random variables are denoted by uppercase letters (X, Y, T) and their realisations by lowercase (x, y, t).

C.2 Linear Algebra

The following symbols are used throughout.

- $\|v\|$: the Euclidean (ℓ^2) norm of vector $v \in \mathbb{R}^d$.
- $\|v\|_p$: the ℓ^p norm, $(\sum_i |v_i|^p)^{1/p}$.
- $\|A\|_F$: the Frobenius norm of matrix A, $\sqrt{\mathrm{tr}(A^\top A)}$.
- $\|A\|_{\mathrm{op}}$: the operator (spectral) norm, $\sigma_{\max}(A)$.
- $\langle u, v \rangle$: the standard inner product on $\mathbb{R}^d$, $u^\top v = \sum_i u_i v_i$.
- $A^\top$: the transpose of matrix A.
- I_d: the $d \times d$ identity matrix.
- $\mathrm{tr}(A)$: the trace of matrix A, $\sum_i A_{ii}$.
- $\mathrm{rank}(A)$: the rank of matrix A.
- $\sigma_i(A)$: the i-th singular value of A (in decreasing order).
- $A \otimes B$: the Kronecker product of matrices A and B.
- $O(d)$: the orthogonal group, $\{Q \in \mathbb{R}^{d \times d} : Q^\top Q = I_d\}$.
- S^{d-1}: the $(d-1)$-dimensional unit sphere, $\{v \in \mathbb{R}^d : \|v\| = 1\}$.

C.3 Probability and Statistics

- $\mathbb{P}[A]$: the probability of event A.
- $\mathbb{E}[X]$: the expectation of random variable X.
- $\mathrm{Var}[X]$: the variance of X, $\mathbb{E}[X^2] - (\mathbb{E}[X])^2$.
- $\mathrm{Cov}[X, Y]$: the covariance of X and Y, $\mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y]$.
- $X \sim D$: the random variable X is distributed according to D.
- $p(x)$: the probability density (or mass) function evaluated at x.
- $p(x|y)$: the conditional density of X given $Y = y$.
- $\mathcal{N}(\mu, \Sigma)$: the Gaussian distribution with mean μ and covariance Σ.

C.4 Information Theory

- $H(X)$: the Shannon entropy of X, $-\sum_x p(x) \log p(x)$ (discrete case).
- $h(X)$: the differential entropy of X, $-\int p(x) \log p(x)\, dx$ (continuous case).
- $I(X;Y)$: the mutual information, $\mathbb{E}\left[\log \frac{p(X,Y)}{p(X)p(Y)}\right]$.
- $I(X;Y|Z)$: the conditional mutual information of X and Y given Z.
- $D_{\mathrm{KL}}(P\|Q)$: the KL divergence from Q to P, $\mathbb{E}_P\left[\log \frac{p(X)}{q(X)}\right]$.
- $\mathrm{TV}(P, Q)$: the total variation distance, $\sup_A |P(A) - Q(A)|$.
- $R(D)$: the rate-distortion function at distortion level D (see Chapter 4).
- $\mathrm{lse}(\beta, z)$: the log-sum-exp function, $\frac{1}{\beta} \log \sum_i e^{\beta z_i}$.

C.5 Frame Theory

- $F = [f_1 \mid \cdots \mid f_m] \in \mathbb{R}^{d \times m}$: a frame matrix.
- $G = F^\top F \in \mathbb{R}^{m \times m}$: the Gram matrix, with $G_{ij} = \langle f_i,\, f_j \rangle$.
- $S = FF^\top \in \mathbb{R}^{d \times d}$: the frame operator.
- $\mu(F) = \max_{i \neq j} |\langle f_i,\, f_j \rangle|$: the coherence.
- δ_k: the Restricted Isometry constant of order k (Definition 1.3).
- $\mathcal{E}(F) = \sum_{i \neq j} |\langle f_i,\, f_j \rangle|^2$: the frame interference energy (Chapter 6).
- UNTF: unit-norm tight frame; $\|f_i\| = 1$ for all i and $FF^\top = (m/d) I_d$.
- ETF: equiangular tight frame; a UNTF with all pairwise coherences equal in magnitude.

C.6 Neural Network Architecture

- $\mathcal{N} = (V, E, W, \phi)$: a feedforward neural network (Definition 2.1).
- V_ℓ: the neurons at layer ℓ.
- $f_{\mathcal{N}} : \mathbb{R}^{|V_0|} \to \mathbb{R}^{|V_L|}$: the function computed by $\mathcal{N}$.
- $\mathcal{C}$: a circuit in $\mathcal{N}$ (Definition 2.1).
- $|E^{\mathcal{C}}|$: the number of edges (size) of circuit $\mathcal{C}$.
- $V_{\text{in}}^{\mathcal{C}}$, $V_{\text{out}}^{\mathcal{C}}$: the input and output node sets of $\mathcal{C}$.
- d_k, d_v: key-query dimension and value dimension in the attention mechanism.
- T: sequence length (number of tokens) in a transformer.
- H: number of attention heads.
- $W_Q^h, W_K^h, W_V^h, W_O^h$: query, key, value, and output weight matrices of head h.

C.7 Transformer Attention

- $A^h(x) = \text{softmax}\Big(xW_Q^h(xW_K^h)^\top/\sqrt{d_k}\Big) \in \mathbb{R}^{T\times T}$: the attention matrix of head h.
- $\alpha_{t,s}$: the attention weight from position t to position s.
- $c_Q(u,v) = \left\|W_O^u W_Q^v\right\|_F$: composition score (query).
- $c_K(u,v) = \|W_O^u W_K^v\|_F$: composition score (key).
- $c_V(u,v) = \|W_O^u W_V^v\|_F$: composition score (value).
- $\mathcal{H}_h(q) = -\sum_s \alpha_s \log \alpha_s$: attention entropy of head h on query q.

C.8 Interpretability Framework

- $h(x) = \sum_i a_i(x) f_i$: the representation of input x.
- $a_i(x)$: the activation of feature i on input x.
- $\lambda_i = I(f_i^\top T; Y)$: the importance of feature i.
- $P(t) = \|W_t[\mathcal{C}_{\text{alg}}]\|_F^2 \,/\, \|W_t\|_F^2$: the algorithmic progress measure.
- $R_{\mathcal{C}}(x) = f_{\mathcal{N}}(x)[V_{\text{out}}] - f_{\mathcal{C}}(x_{\text{in}})$: the circuit residual.
- $\text{MI}(e) = I(T^{(\ell_e)}; Y) - I(T_{-e}^{(\ell_e)}; Y)$: the marginal MI of edge e.
- $K(\mathcal{C})$: the Kolmogorov complexity of circuit $\mathcal{C}$.
- $\text{IPF}(\mathcal{N}, \mathcal{T})$: the interpretability–performance frontier.

C.9 Asymptotics and Order Notation

- $f = O(g)$: there exists $C > 0$ such that $|f(n)| \leq C|g(n)|$ for all large n.
- $f = \Omega(g)$: there exists $c > 0$ such that $|f(n)| \geq c|g(n)|$ for all large n.
- $f = \Theta(g)$: both $f = O(g)$ and $f = \Omega(g)$.
- $f \asymp g$: $f = \Theta(g)$ (used when the implicit constants are not tracked).
- $f \lesssim g$: $f = O(g)$ (used informally when constants are unimportant).
- $a \ll b$: a is much smaller than b in the context at hand.

C.10 Scaling Law Parameters

- N: total number of parameters of a language model.
- D: number of training tokens.
- α: parameter scaling exponent in $\mathcal{L}(N) \sim N^{-\alpha}$.
- β: data scaling exponent in $\mathcal{L}(D) \sim D^{-\beta}$.
- γ: exponent of the feature importance power law, $\lambda_i \sim i^{-\gamma}$.
- $\mathcal{L}_\infty$: the irreducible entropy floor, $\lim_{N,D\to\infty} \mathcal{L}(N, D)$.
- A, B: prefactors in the Chinchilla scaling law (9.1).

C.11 On Mathematical Rigour

Several of the results in this monograph are stated as theorems but proved with varying degrees of rigour. We briefly classify them.

(i) *Rigorous proofs*: Theorems 1.1, 1.4, 1.5, 5.2, Lemmas 1.6, 6.2, 3.2, 4.1, Propositions 4.7, 1.8, 6.6.

(ii) *Proofs with standard techniques, details omitted*: Theorems 2.7, 2.8, 8.1, 8.3, 8.9.

(iii) *Heuristic arguments or sketch proofs*: Theorems 3.3 (Kramers formula applied heuristically), 9.2 (optimal interference allocation assumed), 7.4 (information bottleneck optimality assumed).

(iv) *Hypotheses and conjectures*: Hypothesis 7.3, Hypothesis 10.9, Corollary 6.7.

Category (iii) results are presented as theorems because we believe the heuristic arguments are essentially correct and can be made rigorous with additional technical work. The open problems in Appendix B identify the specific gaps. Category (iv) results are presented as hypotheses because they are plausible on the basis of the theory but not yet verified by sufficiently rigorous argument or sufficiently broad empirical confirmation.

C.12 Glossary of Technical Terms

This section provides brief definitions of technical terms used in the monograph, ordered alphabetically for reference.

Absolute bound
The inequality $m \leq d(d+1)/2$ (real) or $m \leq d^2$ (complex) that is necessary for the existence of an equiangular tight frame with m vectors in dimension d.

Activation patching
An interpretability technique in which the activation of a specific node is replaced by its value on a different input, measuring the causal effect on the output.

Algorithmic progress measure
The quantity $P(t) = \|W_t[\mathcal{C}_{\text{alg}}]\|_F^2 \,/\, \|W_t\|_F^2$, measuring the fraction of the total weight norm in the algorithmic circuit during training (§3.3).

Attention entropy
The entropy $\mathcal{H}_h(q) = -\sum_s \alpha_s \log \alpha_s$ of the attention distribution of head h on query q; measures the diffuseness of attention (§5.7).

Bayesian risk
The expected loss of a decision rule under a prior distribution; the rate-distortion function is a Bayesian risk minimisation.

Bottleneck representation
The optimal representation T^* minimising the IB functional $I(X;T) - \beta I(T;Y)$ for a given tradeoff β (§4.3).

Capacity-interference tradeoff
The fundamental tradeoff between the number of representable features m and the pairwise interference ε: $\log m \lesssim \varepsilon^2 d$ (Proposition 1.7).

Causal faithfulness
A feature is causally faithful for a property if intervening on the feature changes the property (Definition 10.5); the condition for representation steering to reliably control model behaviour.

Chinchilla scaling laws
The empirical power laws $\mathcal{L}(N, D) \approx \mathcal{L}_\infty + AN^{-\alpha} + BD^{-\beta}$ with $\alpha \approx 0.336$, $\beta \approx 0.283$ [31].

Circuit
A subgraph of a neural network's computational graph that implements an identifiable function (Definition 2.1).

Circuit crystallisation
The process by which a latent algorithmic circuit becomes the dominant computation in a network, typically occurring rapidly (Stage 3 of grokking, §3.7).

Circuit discovery
The algorithmic problem of finding the smallest faithful circuit in a network (equation 2.2); Σ_2^P-complete (Theorem 8.3).

Circuit isomorphism
A structural equivalence between two circuits related by orthogonal transformations of layer representations and overall scaling (Definition 7.1).

Circuit residual
The difference between the network's output and the circuit's output, decomposable into task-correlated and noise components (§2.5).

Coherence
The maximum absolute inner product between distinct feature vectors: $\mu(F) = \max_{i \neq j} |\langle f_i, f_j \rangle|$.

Composition score
A measure of the information flow between two attention heads; the Frobenius norm of the product of their output and input weight matrices (Definition 2.6).

Compressed sensing
The field studying the recovery of sparse signals from underdetermined linear measurements; the mathematical foundation of the superposition hypothesis.

Cross-polytope
The convex hull of $\{\pm e_1, \ldots, \pm e_d\}$ in $\mathbb{R}^d$ (a higher-dimensional octahedron); the optimal frame configuration for $m = 2d$ features in $\mathbb{R}^d$.

Deceptive circuit
A circuit pattern in which a network implements a hidden objective alongside its apparent task (Definition 10.1).

Diffuse attention
The regime $\mathcal{H}_h(q) \approx \log T$ (near-uniform attention distribution); implements aggregation over all source positions.

Double descent
The non-monotone behaviour of test loss as model size increases through the interpolation threshold; explained as circuit competition (Proposition 9.7).

Emergence
The abrupt appearance of a capability at a critical model scale; predicted by Theorem 9.5 and explained as a circuit formation threshold.

Equiangular tight frame (ETF)
A unit-norm tight frame with all pairwise inner products equal in magnitude; achieves the Welch bound (Theorem 1.1).

Feature
A unit vector in a representation space encoding a property of the input; the representational atom of the mechanistic interpretability framework.

Feature importance
The mutual information $\lambda_i = I(f_i^\top T; Y)$ between the activation of feature i and the label; follows a power law in natural language.

Feature splitting
The phase transition from a polysemantic configuration (two concepts sharing a feature direction) to a monosemantic configuration (two orthogonal directions); triggered by increasing sparsity (Theorem 6.4).

Faithful circuit
A circuit whose output approximates the full network's output to within ε in L^2 loss (Definition 2.4).

Frame
A collection of vectors in $\mathbb{R}^d$ (possibly more than d) satisfying a stability condition; the mathematical language for the feature representation (§1.2).

Frame interference energy
The functional $\mathcal{E}(F) = \sum_{i \neq j} |\langle f_i, f_j \rangle|^2$ measuring the total squared interference between features; minimised by UNTFs.

Free energy barrier
The difference between the free energy of the transition state and the current basin; controls the escape time via Kramers' formula (§3.5).

Grokking
The phenomenon of delayed generalisation: a network first memorises the training data and later, through continued training, learns to generalise (Definition 3.1).

Heteroassociative memory
A content-addressable memory that matches queries against one set of patterns (keys) but retrieves from another (values); the structure implemented by transformer attention (Proposition 5.3).

Hidden progress measure
See algorithmic progress measure.

Hopfield network
A neural network implementing content-addressable associative memory; the modern version (Theorem 5.2) has exponential storage capacity.

In-context learning
The ability of large language models to learn new tasks from examples in the prompt; implemented by induction head circuits (§3.8).

Induction head
An attention head that, given a repeated pattern $[A, B, \ldots, A]$, predicts B as the next token; a key mechanism for in-context learning [49].

Information bottleneck
The framework minimising $I(X;T) - \beta I(T;Y)$ (compression vs. relevance); the optimal faithful circuit solves the IB problem (Theorem 4.2).

Interaction graph
The graph of information flow between attention heads and MLP layers in a transformer (Definition 2.6).

Interpretability criterion
The triple of conditions (faithfulness, low Kolmogorov complexity, incoherent frame) that a circuit must satisfy to be interpretable (Corollary 4.8).

Interpretability–performance frontier
The Pareto frontier of achievable (test error, circuit complexity) pairs (Definition 8.8).

Kolmogorov complexity
The length of the shortest program producing a description of an object; used to measure the complexity of a circuit $K(\mathcal{C})$.

Kramers' formula
The formula for the mean exit time from a potential well: $T_{\text{escape}} \propto \exp(\Delta\mathcal{F}/T_{\text{eff}})$ [30].

Logical depth
The minimum computation time of a fast program for an object; a measure of the organised complexity of a circuit (§8.8, [9]).

MDL bound
The bound $|E^{\mathcal{C}^*}| \leq K(f_Y)/\log(1/\delta)$ on the size of the minimal description length circuit (Proposition 4.7).

Mechanistic interpretability
The programme of reverse-engineering neural networks at the level of individual computations, identifying circuits and features.

Mechanistically irreducible circuit
A circuit that cannot be replaced by any simpler faithful sub-circuit (Definition 8.6).

MINE
Mutual Information Neural Estimation; a variational lower bound on mutual information [6].

Modern Hopfield network
A Hopfield network with a non-linear energy function achieving exponential storage capacity (Theorem 5.2).

Modularity hypothesis
The claim that neural networks implement their computations through identifiable, reusable functional units (circuits).

Mutual information attribution
The marginal mutual information $\text{MI}(e)$ of an edge, measuring the information contribution of removing that edge (§4.6).

Norm gap
The difference $\|W_{\text{mem}}\|_F^2 - \|W_{\text{alg}}\|_F^2 \sim N - s \gg 0$ between the weight norms of the memorisation and algorithmic solutions (Lemma 3.2).

Order parameter
The quantity $\phi = |\rho|$ measuring the overlap between feature directions; equals 1 in the merged (polysemantic) phase and decreases in the split (monosemantic) phase (§6.6).

Parallel composition
The operation $\mathcal{C}_1 \oplus \mathcal{C}_2$ combining two circuits over disjoint node sets (Definition 2.3).

Phase transition
A qualitative change in the state of a system at a critical value of a control parameter; grokking (Chapter 3) and feature splitting (Chapter 6) are phase transitions.

Platonic circuit
The unique (up to isomorphism) minimal faithful circuit achieving the information bottleneck optimum for a task (Hypothesis 7.3).

Platonic Representation Hypothesis
The claim that networks trained on the same task converge to circuits in the same universality class (Hypothesis 7.3).

Polysemanticity
The phenomenon of individual neurons responding to multiple semantically unrelated stimuli; a consequence of superposition.

Population code
A distributed representation in which a stimulus is encoded by the joint activity of multiple neurons; the neuroscience analogue of superposition.

Rate-distortion function
The minimum mutual information achievable at a given distortion level: $R(D) = \min_{p(T|X):\mathbb{E}[d(Y,\hat{Y})]\leq D} I($ (§4.2).

Representation similarity analysis
A method for comparing representations across networks or brain areas using representational dissimilarity matrices (§7.8).

Residual stream
The running sum of contributions from all layers in a transformer; the shared communication bus for attention heads and MLP blocks.

Restricted Isometry Property (RIP)
The condition that a frame matrix acts almost isometrically on sparse vectors (Definition 1.3); guarantees exact recovery.

Scaling law
The empirical power law $\mathcal{L}(N, D) \approx \mathcal{L}_\infty + AN^{-\alpha} + BD^{-\beta}$ for neural network test loss; derived from the superposition framework (Theorem 9.2).

Sequential composition
The operation $\mathcal{C}_2 \circ \mathcal{C}_1$ chaining two circuits head-to-tail (Definition 2.2).

Sharp attention
The regime $\mathcal{H}_h(q) \approx 0$ (highly concentrated attention); implements a winner-take-all lookup (§5.7).

Sparse coding
The representation of signals as sparse linear combinations of basis functions; the neuroscience precursor to the superposition hypothesis.

Sparsity
The property of activation vectors with few non-zero entries; the key structural assumption that makes superposition feasible.

Sufficient statistic
A feature f_i is δ-sufficient for property g_i if it captures all mutual information about g_i up to δ (Definition 4.3).

Superposition
The encoding of multiple features as directions in a shared representation space, using sparsity to manage interference; the central phenomenon studied in Chapter 1.

Symmetry breaking
The transition from a symmetric (merged) phase to a broken-symmetry (split) phase; governs feature splitting via the Ising/Potts universality class (§6.6).

Transformer circuit
A faithful subgraph of a transformer's interaction graph (Definition 2.6).

Universality class
An equivalence class of circuits under $(0, 0)$-isomorphism; classifies qualitatively distinct circuit architectures (§7.2).

Unit-norm tight frame (UNTF)
A frame with unit-norm vectors and tight frame condition $FF^\top = (m/d)I_d$; the global minimiser of the frame interference energy (Lemma 6.2).

VIB
Variational Information Bottleneck; a tractable approximation to the information bottleneck objective using variational bounds (§4.7).

Welch bound
The lower bound $\mu(F) \geq \sqrt{(m-d)/[d(m-1)]}$ on the coherence of any unit-norm frame (Theorem 1.1).

Weight decay
ℓ^2 regularisation on the network weights; the thermodynamic force driving grokking by penalising high-norm solutions.

Zipf's law
The empirical power law $p_i \propto i^{-s}$ for the frequency of the i-th most common word in natural language; the origin of the power-law feature importance distribution (Proposition 9.1).

Appendix D

Annotated Bibliography

This appendix provides an annotated guide to the most important references for each chapter, with brief comments on their content and their relation to the theory developed in this monograph. It is intended as a reading guide for researchers entering the field, not as a comprehensive survey of the literature.

D.1 Frame Theory and Compressed Sensing (Chapter 1)

Christensen, O. (2016). *An Introduction to Frames and Riesz Bases*. Birkhäuser. *The standard reference for frame theory, covering the Welch bound, tight frames, ETFs, and their applications to signal processing. Chapter 1 of this monograph draws heavily on Chapters 5–7 of Christensen.*

Welch, L. R. (1974). "Lower Bounds on the Maximum Cross Correlation of Signals." *IEEE Transactions on Information Theory* 20(3):397–399. *The original paper establishing the Welch bound. The two-page proof is essentially the one in Theorem 1.1; we have expanded it slightly for pedagogical clarity.*

Candès, E. J., Romberg, J., and Tao, T. (2006). "Robust Uncertainty Principles: Exact Signal Reconstruction from Highly Incomplete Frequency Information." *IEEE Transactions on Information Theory* 52(2):489–509. *The foundational paper of compressed sensing, establishing the ℓ^1 recovery theorem (Theorem 1.4) for random frames. The RIP formulation and the Candès-Romberg-Tao theorem are central to Chapter 1's capacity analysis.*

Donoho, D. L. (2006). "Compressed Sensing." *IEEE Transactions on Information Theory* 52(4):1289–1306. *Concurrent and complementary to the Candès-Romberg-Tao work; establishes compressed sensing for deterministic matrices satisfying RIP. The sparsity-based recovery criterion of §1.3 draws from this paper.*

Ledoux, M. (2001). *The Concentration of Measure Phenomenon*. AMS. *The definitive mathematical treatment of concentration inequalities, including the sphere concentration inequality (Lemma A.2) and the Johnson-Lindenstrauss flattening lemma. Chapter 1's exponential capacity argument uses these tools.*

Elhage, N. et al. (2022). "Toy Models of Superposition." Transformer Circuits Thread. *The paper that introduced the superposition hypothesis to the mechanistic interpretability community. Demonstrates empirically that neural networks encode features in superposition when features outnumber neurons, and identifies the polytope configurations predicted by Proposition 6.6.*

Casazza, P. G. and Kutyniok, G. (2013). *Finite Frames: Theory and Applications*. Birkhäuser. *The standard reference for finite frame theory, including ETFs and their con-*

nection to graph theory. The classification of optimal configurations in Proposition 6.6 draws from this volume.

D.2 Circuit Theory (Chapter 2)

Elhage, N. et al. (2021). "A Mathematical Framework for Transformer Circuits." Transformer Circuits Thread. *The foundational paper for the transformer circuit approach. Introduces the residual stream, attention heads, and composition scores; provides the first formal account of how information flows through a multi-layer transformer. Definition 2.6 is adapted from this paper.*

Wang, K. et al. (2022). "Interpretability in the Wild: A Circuit for Indirect Object Identification in GPT-2 Small." arXiv:2211.00593. *The definitive circuit analysis paper, identifying the IOI circuit with the precision demanded by Definition 2.1. Introduces activation patching and path patching as circuit verification tools. The worked example of §2.8 is based on this paper.*

Eldan, R. and Shamir, O. (2016). "The Power of Depth for Feedforward Neural Networks." COLT. *Proves the depth-width tradeoff for neural networks: functions approximable by small deep networks require exponentially large shallow networks. The basis for Theorem 2.7 and the circuit complexity analysis of §2.6.*

Fong, B. and Spivak, D. I. (2019). *An Invitation to Applied Category Theory.* Cambridge University Press. *Provides the categorical framework for circuit composition (§2.3). The symmetric monoidal category structure of circuits is a special case of the string diagram calculus developed in this book.*

D.3 Grokking and Phase Transitions (Chapter 3)

Power, A. et al. (2022). "Grokking: Generalisation Beyond Overfitting on Small Algorithmic Datasets." arXiv:2201.02177. *The paper that named and characterised grokking. Discovered that generalisation can occur thousands of gradient steps after memorisation, across a range of algorithmic tasks. Definition 3.1 captures the phenomenology they describe.*

Nanda, N. et al. (2023). "Progress Measures for Grokking via Mechanistic Interpretability." arXiv:2301.05217. *The mechanistic analysis of grokking in modular addition. Identified the Fourier algorithm of §3.4 and the progress measure $P(t)$ of* (3.1)*. Our four-stage dynamical picture (§3.7) is a formalisation of their empirical observations.*

Barak, B. et al. (2022). "Hidden Progress in Deep Learning: SGD Learns Parities Near the Computational Limit." arXiv:2207.08799. *Studies grokking in the context of sparse parity learning; introduces the "hidden progress" perspective that gradient descent is making algorithmic progress even when the loss is stagnant. Connects grokking to computational lower bounds for learning.*

Hänggi, P., Talkner, P., and Borkovec, M. (1990). "Reaction-Rate Theory: Fifty Years After Kramers." *Reviews of Modern Physics* 62(2):251–341. *The comprehensive review of Kramers' escape rate theory, which provides the thermodynamic framework for the grokking phase transition (Theorem 3.3). The mean exit time formula* (3.5) *is a special case of the general theory reviewed here.*

Olsson, C. et al. (2022). "In-Context Learning and Induction Heads." Transformer Circuits Thread. *Identifies induction heads as the mechanism for in-context learning and*

studies their formation during training. The connection to grokking developed in §3.8 is a novel application of their findings.

D.4 Information Theory of Circuits (Chapter 4)

Tishby, N., Pereira, F. C., and Bialek, W. (1999). "The Information Bottleneck Method." *37th Annual Allerton Conference. The foundational paper of the information bottleneck. Introduces the IB functional* (4.4) *and the characterisation of the optimal bottleneck representation. Chapter 4 connects the IB framework to the circuit formalism of Chapter 2.*

Berger, T. (1971). *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Prentice-Hall. *The classical text on rate-distortion theory, including the lower bound of Lemma 4.1 and the Gaussian channel tightness of Proposition 4.13.*

Grünwald, P. D. (2007). *The Minimum Description Length Principle.* MIT Press. *The definitive treatment of MDL, connecting information theory to statistical model selection. Proposition 4.7 uses the MDL excess risk bounds from §12 of this volume.*

Belghazi, M. I. et al. (2018). "MINE: Mutual Information Neural Estimation." ICML. *Introduces the variational lower bound* (4.17) *for mutual information, enabling tractable estimation of $I(X;T)$ in high dimensions. The practical implementation of the IB framework in §4.7 uses this estimator.*

D.5 Attention Heads as Associative Memories (Chapter 5)

Hopfield, J. J. (1982). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *PNAS* 79(8):2554–2558. *The original paper introducing Hopfield networks as content-addressable memories. The classical capacity bound ($m^* \approx 0.138n$) motivates the need for modern Hopfield networks.*

Amit, D. J., Gutfreund, H., and Sompolinsky, H. (1985). "Storing Infinite Numbers of Patterns in a Spin-Glass Model of Neural Networks." *Physical Review Letters* 55(14):1530–1533. *Uses the replica method from spin glass theory to derive the classical Hopfield capacity bound (Theorem 5.1). The first paper to treat neural network memory as a statistical mechanics problem.*

Ramsauer, H. et al. (2021). "Hopfield Networks Is All You Need." arXiv:2008.02217. *The key paper connecting modern Hopfield networks to transformer attention (Proposition 5.3). Proves the exponential storage capacity (Theorem 5.2) and makes the mathematical equivalence with attention explicit.*

Krotov, D. and Hopfield, J. J. (2016). "Dense Associative Memory for Pattern Recognition." NeurIPS. *Introduces the modern Hopfield network with exponential capacity, predating the Ramsauer et al. paper. Uses a polynomial energy function rather than the log-sum-exp of* (5.2).

D.6 Feature Splitting and Phase Transitions (Chapter 6)

Benedetto, J. J. and Fickus, M. (2003). "Finite Normalized Tight Frames." *Advances in Computational Mathematics* 18:357–385. *Introduces the frame force (Lemma the gradient flow result) and proves that gradient descent on the frame interference energy converges to UNTFs. The basis for the energy landscape analysis of §6.3.*

Wu, F. Y. (1982). "The Potts Model." *Reviews of Modern Physics* 54(1):235–268. *The comprehensive review of the q-state Potts model, whose universality class governs feature splitting for r features (§6.6). The mean-field exponents and symmetry breaking pattern of* (6.6) *are standard results from this model.*

Goldenfeld, N. (1992). *Lectures on Phase Transitions and the Renormalization Group*. Addison-Wesley. *The standard mathematical treatment of Landau theory and renormalisation group methods. The Landau expansion of §6.8 and the Ginzburg criterion for mean-field validity draw from Chapters 4–5 of this text.*

D.7 Universality (Chapter 7)

Olah, C. et al. (2020). "Zoom In: An Introduction to Circuits." Distill. *The paper that introduced circuit motifs and documented their universality across architectures. Identified curve detectors, high-low frequency detectors, and other universal circuit elements in vision models. The convergent evolution metaphor (§7.7) is developed here.*

Huh, M. et al. (2024). "The Platonic Representation Hypothesis." arXiv:2405.07987. *Empirically tests and develops the universality hypothesis, finding that representations of different modalities and architectures converge to a common geometry. The theoretical explanation is the information bottleneck universality of Theorem 7.4.*

Yamins, D. L. K. and DiCarlo, J. J. (2016). "Using Goal-Driven Deep Learning Models to Understand Sensory Cortex." *Nature Neuroscience* 19(3):356–365. *Demonstrates that deep learning models trained on visual tasks converge to representations resembling those of the primate visual cortex. The strongest empirical evidence for cross-domain universality, connecting Chapter 7's theory to neuroscience.*

D.8 Limits of Interpretability (Chapter 8)

Siegelmann, H. T. and Sontag, E. D. (1991). "Turing Computability with Neural Nets." *Applied Mathematics Letters* 4(6):77–80. *Establishes the Turing-completeness of recurrent neural networks, underpinning the undecidability proof of Theorem 8.1. Shows that neural network computation is as powerful as (and as unpredictable as) general computation.*

Katz, G. et al. (2017). "Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks." CAV. *Despite the undecidability of general neural network verification (Theorem 8.1), provides a decision procedure for finite-domain verification of ReLU networks. In practice, verification is feasible for small networks on bounded domains.*

Li, M. and Vitányi, P. (2008). *An Introduction to Kolmogorov Complexity and Its Applications*, 3rd ed. Springer. *The standard reference for Kolmogorov complexity and its applications, including the chain rule* (9.12) *and the MDL connection. Chapters 2 and 7 are most relevant to this monograph.*

Bennett, C. H. (1988). "Logical Depth and Physical Complexity." *The Universal Turing Machine: A Half-Century Survey*. Oxford University Press. *Introduces logical depth as a measure of organised complexity distinct from Kolmogorov complexity. Definition 8.12 uses logical depth to extend the interpretability criterion beyond mere compressibility.*

D.9 Scaling Laws (Chapter 9)

Hoffmann, J. et al. (2022). "Training Compute-Optimal Large Language Models." arXiv:2203.15556. *The Chinchilla paper, establishing the scaling law* (9.1) *and the optimal compute allocation of Proposition 9.8. The empirical exponents* $\alpha \approx 0.336$ *and* $\beta \approx 0.283$ *that Theorem 9.2 derives from first principles.*

Michaud, E. J. et al. (2023). "The Quantization Model of Neural Scaling." arXiv:2303.13506. *Proposes a model of scaling laws based on "quantized" features (quanta) with a discrete importance distribution. Related to but distinct from the power-law feature model of Theorem 9.2; both approaches explain the scaling law exponents.*

Wei, J. et al. (2022). "Emergent Abilities of Large Language Models." *TMLR. Documents the emergence phenomenon (Definition 9.4) across many capabilities and model sizes. The apparent discontinuity is explained by Corollary 9.6 as a measurement artefact from threshold metrics.*

D.10 AI Safety and Alignment (Chapter 10)

Hubinger, E. et al. (2019). "Risks from Learned Optimization in Advanced Machine Learning Systems." arXiv:1906.01820. *The foundational paper on deceptive alignment, introducing the mesa-optimiser framework. Definition 10.1 is a circuit-theoretic formalisation of the deceptive alignment scenario described in this paper.*

Burns, C. et al. (2022). "Discovering Latent Knowledge in Language Models Without Supervision." arXiv:2212.03827. *Introduces Contrast-Consistent Search (CCS) for identifying truthfulness features in language models without labelled data. The empirical motivation for Hypothesis 10.9 and for the truthfulness analysis of §10.4.*

Zou, A. et al. (2023). "Representation Engineering: A Top-Down Approach to AI Transparency." arXiv:2310.01405. *Develops representation engineering as a method for identifying and manipulating feature directions corresponding to high-level concepts (including values). The causal faithfulness criterion (Definition 10.5) formalises when representation engineering reliably steers behaviour.*

Christiano, P. et al. (2017). "Deep Reinforcement Learning from Human Preferences." NeurIPS. *Introduces reinforcement learning from human feedback (RLHF), the training paradigm most relevant to the value alignment discussion of §10.7. The scalable oversight framework of §10.6 is motivated by the limitations of RLHF identified in this paper.*

Pearl, J. (2009). *Causality: Models, Reasoning, and Inference*, 2nd ed. Cambridge University Press. *The mathematical framework for causal inference, including the do-calculus used in Definition 10.5. Chapter 3 of Pearl is essential background for the causal faithfulness analysis of §10.4.*

Bibliography

[1] Shun-ichi Amari. *Information Geometry and Its Applications*. Springer, Tokyo, 2016.

[2] Daniel J. Amit, Hanoch Gutfreund, and Haim Sompolinsky. Storing infinite numbers of patterns in a spin-glass model of neural networks. *Physical Review Letters*, 55(14): 1530–1533, 1985.

[3] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. arXiv preprint arXiv:1606.06565, 2016.

[4] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[5] Boaz Barak, Benjamin L. Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: SGD learns parities near the computational limit. arXiv preprint arXiv:2207.08799, 2022.

[6] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. MINE: Mutual information neural estimation. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 531–540, 2018.

[7] Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine learning practice and the bias-variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

[8] John J. Benedetto and Matthew Fickus. Finite normalized tight frames. *Advances in Computational Mathematics*, 18:357–385, 2003.

[9] Charles H. Bennett. Logical depth and physical complexity. *The Universal Turing Machine: A Half-Century Survey*, pages 227–257, 1988.

[10] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, New York, 2nd edition, 1985.

[11] Toby Berger. *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[12] Lenore Blum, Mike Shub, and Steve Smale. On a theory of computation and complexity over the real numbers. *Bulletin of the American Mathematical Society*, 21(1):1–46, 1989.

[13] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. *Algebraic Complexity Theory*. Springer, Berlin, 1997.

[14] Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. arXiv preprint arXiv:2212.03827, 2022.

[15] Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[16] Peter G. Casazza and Gitta Kutyniok. *Finite Frames: Theory and Applications*. Birkhäuser, Basel, 2013.

[17] Ole Christensen. *An Introduction to Frames and Riesz Bases*. Birkhäuser, Basel, 2nd edition, 2016.

[18] Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[19] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. arXiv preprint arXiv:1906.04341, 2019.

[20] Henry Cohn and Noam Elkies. New upper bounds on sphere packings I. *Annals of Mathematics*, 157(2):689–714, 2003.

[21] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52 (4):1289–1306, 2006.

[22] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. *Conference on Learning Theory (COLT)*, pages 907–940, 2016.

[23] Nelson Elhage, Neel Nanda, Catherine Olsson, et al. A mathematical framework for transformer circuits. Transformer Circuits Thread, 2021. `https://transformer-circuits.pub/2021/framework/index.html`.

[24] Nelson Elhage et al. Toy models of superposition. Transformer Circuits Thread, 2022. `https://transformer-circuits.pub/2022/toy_model/index.html`.

[25] Matthew Fickus and Dustin G. Mixon. Tables of the existence of equiangular tight frames. *arXiv preprint arXiv:1504.00253*, 2015.

[26] Brendan Fong and David I. Spivak. *An Invitation to Applied Category Theory*. Cambridge University Press, 2019.

[27] Nigel Goldenfeld. *Lectures on Phase Transitions and the Renormalization Group*. Addison-Wesley, Reading, MA, 1992.

[28] Peter Grünwald and A. Philip Dawid. Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory. *The Annals of Statistics*, 32(4):1367–1433, 2004.

[29] Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, Cambridge, MA, 2007.

[30] Peter Hänggi, Peter Talkner, and Michal Borkovec. Reaction-rate theory: Fifty years after Kramers. *Reviews of Modern Physics*, 62(2):251–341, 1990.

[31] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, et al. Training compute-optimal large language models. arXiv preprint arXiv:2203.15556, 2022.

[32] John J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[33] Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. arXiv preprint arXiv:1906.01820, 2019.

[34] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. The platonic representation hypothesis. arXiv preprint arXiv:2405.07987, 2024.

[35] William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in Modern Analysis and Probability*, volume 26 of *Contemporary Mathematics*, pages 189–206. AMS, 1984.

[36] Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Computer Aided Verification (CAV)*, pages 97–117, 2017.

[37] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB)*, pages 180–191, 2004.

[38] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis — connecting the branches of systems neuroscience. Frontiers in Systems Neuroscience, 2008.

[39] Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, 2016.

[40] Michel Ledoux. *The Concentration of Measure Phenomenon*. American Mathematical Society, Providence, RI, 2001.

[41] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, New York, 3rd edition, 2008.

[42] David Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, San Francisco, 1982.

[43] Eric J. Michaud, Ziming Liu, Uzay Girit, and Max Tegmark. The quantization model of neural scaling. arXiv preprint arXiv:2303.13506, 2023.

[44] Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations (ICLR)*, 2021.

[45] Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. arXiv preprint arXiv:2301.05217, 2023.

[46] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. Distill, 2020. `https://distill.pub/2020/circuits/zoom-in/`.

[47] Enzo Olivieri and Maria Eulália Vares. *Large Deviations and Metastability*. Cambridge University Press, 2005.

[48] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1996.

[49] Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. In-context learning and induction heads. Transformer Circuits Thread, 2022. `https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html`.

[50] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

[51] Steven T. Piantadosi. Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130, 2014.

[52] Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. arXiv preprint arXiv:2201.02177, 2022.

[53] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Thomas Adler, Lukas Gruber, Markus Holzleitner, Milena Pavlović, Geir Kjetil Sandve, Victor Greiff, David Kreil, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. Hopfield networks is all you need. arXiv preprint arXiv:2008.02217, 2021.

[54] Mark Rudelson and Roman Vershynin. Hanson-Wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18:1–9, 2013.

[55] Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. Are emergent abilities of large language models a mirage? arXiv preprint arXiv:2304.15004, 2023.

[56] H. Sebastian Seung and Haim Sompolinsky. Simple models for reading neuronal population codes. *Proceedings of the National Academy of Sciences*, 90(22):10749–10753, 1993.

[57] Hava T. Siegelmann and Eduardo D. Sontag. Turing computability with neural nets. *Applied Mathematics Letters*, 4(6):77–80, 1991.

[58] Michael Strevens. *Depth: An Account of Scientific Explanation*. Harvard University Press, 2008.

[59] Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. Extracting latent steering vectors from pretrained language models. arXiv preprint arXiv:2205.05124, 2022.

[60] Matus Telgarsky. Benefits of depth in neural networks. In *Conference on Learning Theory (COLT)*, pages 1517–1539, 2016.

[61] Naftali Tishby and Ravid Schwartz-Ziv. Opening the black box of deep neural networks via information. arXiv preprint arXiv:1703.00810, 2017.

[62] Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. *37th Annual Allerton Conference on Communication, Control, and Computing*, pages 368–377, 1999.

[63] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Jose Vazquez, Ulisse Mini, and Monte MacDiarmid. Activation addition: Steering language models without optimization. arXiv preprint arXiv:2308.10248, 2023.

[64] Kevin Wang, Luc Alexandre, et al. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. arXiv preprint arXiv:2211.00593, 2022.

[65] Jason Wei, Yi Tay, Rishi Bommasani, et al. Emergent abilities of large language models. Transactions on Machine Learning Research, 2022.

[66] L. R. Welch. Lower bounds on the maximum cross correlation of signals. *IEEE Transactions on Information Theory*, 20(3):397–399, 1974.

[67] Stephen Wolfram. *A New Kind of Science*. Wolfram Media, Champaign, IL, 2002.

[68] F. Y. Wu. The Potts model. *Reviews of Modern Physics*, 54(1):235–268, 1982.

[69] Daniel L. K. Yamins and James J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience*, 19(3):356–365, 2016.

[70] Andy Zou, Long Phan, Sarah Chen, et al. Representation engineering: A top-down approach to AI transparency. arXiv preprint arXiv:2310.01405, 2023.